ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS E INFORMÁTICA

AUTOR: SALGADO YÁNEZ ANGEL LENIN

TEMA: ANÁLISIS DE LAS APLICACIONES WEB DE LA SUPERINTENDENCIA DE BANCOS Y SEGUROS, UTILIZANDO LAS RECOMENDACIONES TOP TEN DE OWASP PARA DETERMINAR LOS RIESGOS MÁS CRÍTICOS DE SEGURIDAD E IMPLEMENTAR BUENAS PRÁCTICAS DE SEGURIDAD PARA EL DESARROLLO DE SUS APLICATIVOS

DIRECTOR: ING. RON MARIO MSC.

CODIRECTOR: ING. SOLIS FERNANDO MSC.

SANGOLQUÍ, ABRIL 2014

CERTIFICADO

Ing. Mario Ron. MSc

Ing. Fernando Solís. MSc

CERTIFICAN

Que el trabajo titulado “ANÁLISIS DE LAS APLICACIONES WEB DE LA SUPERINTENDENCIA DE BANCOS Y SEGUROS, UTILIZANDO LAS RECOMENDACIONES TOP TEN DE OWASP PARA DETERMINAR LOS RIESGOS MÁS CRÍTICOS DE SEGURIDAD E IMPLEMENTAR BUENAS PRÁCTICAS DE SEGURIDAD PARA EL DESARROLLO DE SUS APLICATIVOS” realizado por el Sr. SALGADO YÁNEZ ANGEL LENIN, ha sido guiado y revisado periódicamente y cumple normas estatuarias establecidas por la Universidad de las Fuerzas Armadas “ESPE”.

Sangolquí, abril del 2014

Ing. Mario Ron MSc	Ing. Fernando Solís MSc
DIRECTOR	CODIRECTOR

DECLARACIÓN DE RESPONSABILIDAD

SALGADO YÁNEZ ANGEL LENIN

DECLARO QUE:

El proyecto de grado denominado “ANÁLISIS DE LAS APLICACIONES WEB DE LA SUPERINTENDENCIA DE BANCOS Y SEGUROS, UTILIZANDO LAS RECOMENDACIONES TOP TEN DE OWASP PARA DETERMINAR LOS RIESGOS MÁS CRÍTICOS DE SEGURIDAD E IMPLEMENTAR BUENAS PRÁCTICAS DE SEGURIDAD PARA EL DESARROLLO DE SUS APLICATIVOS”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme a las fuentes que se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, abril del 2014

_____

Salgado Yánez Angel Lenin

AUTORIZACIÓN

Yo, SALGADO YÁNEZ ANGEL LENIN

Autorizo a la UNIVERSIDAD DE LA FUERZAS ARMADAS “ESPE”, la publicación, en la biblioteca virtual de la Institución del trabajo “ANÁLISIS DE LAS APLICACIONES WEB DE LA SUPERINTENDENCIA DE BANCOS Y SEGUROS, UTILIZANDO LAS RECOMENDACIONES TOP TEN DE OWASP PARA DETERMINAR LOS RIESGOS MÁS CRÍTICOS DE SEGURIDAD E IMPLEMENTAR BUENAS PRÁCTICAS DE SEGURIDAD PARA EL DESARROLLO DE SUS APLICATIVOS”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, abril del 2014

_____

Salgado Yánez Angel Lenin

DEDICATORIA

Dedico esta tesis primeramente a mi Padre Ángel Salgado C., por su apoyo incondicional como padre y amigo, ya que gracias a él he cumplido mis metas trazadas y es un ejemplo de superación constante que me inspira para seguir creciendo profesionalmente y como ser humano.

A mi madre, Inés Yánez quien con su eterno amor y comprensión me supo guiar por el camino del bien y estar conmigo en los momentos más adversos de mi vida.

A mi hermana, Mónica Salgado y a la vez como una madre también, gracias por su apoyo en todas las etapas de mi vida, por ser un ejemplo e influir madurez en mi para cumplir con mis objetivos planteados.

Lenin Salgado

AGRADECIMIENTOS

A toda mi familia, por estar siempre conmigo y por sus buenos consejos los mismos que fueron un aliento para continuar con mis estudios.

A Tatiana Salas, por su paciencia, comprensión y bondad, gracias por estar siempre a mi lado para alcanzar mis sueños.

A mis directores de tesis Ing. Mario Ron e Ing. Fernando Solís por guiarme y apoyarme en todo momento en el desarrollo de este proyecto.

A la Superintendencia de Bancos y Seguros por brindarme la oportunidad de ser parte de su equipo de trabajo y apoyarme con todas las herramientas tecnológicas necesarias.

A mis compañeros y amigos de trabajo, gracias por transmitirme todos sus conocimientos y experiencia de manera desinteresada.

A la Universidad de las Fuerzas Armadas – ESPE, mi segundo hogar, quien me formo profesionalmente.

Lenin Salgado

ÍNDICE DE CONTENIDOS

CERTIFICADO i
DECLARACIÓN DE RESPONSABILIDAD ii
AUTORIZACIÓN iii
DEDICATORIA iv
AGRADECIMIENTOS v
ÍNDICE DE TABLAS ix
ÍNDICE DE FIGURAS x
RESUMEN xi
ABSTRACT xii
GLOSARIO DE NOMENCLATURAS xiii
CAPÍTULO 1 1
1.1. Antecedentes 4
1.2. Justificación 13
1.3. Objetivos 13
1.3.1. Objetivo General 13
1.3.2. Objetivo Específicos 13
1.4. Alcance o Meta 14
1.5. Metodología de Investigación 14
CAPÍTULO 2 17
MARCO TEÓRICO 17
2.1. Introducción a las aplicaciones web y la seguridad 18
2.1.1 Cliente – Servidor 19
2.1.2 Protocolo HTTP y HTTPS 20
2.2. Aplicaciones Web 22
2.2.1. Evolución de las Aplicaciones web 22
2.2.2. Arquitectura en capas. 26
2.3. Seguridad en las aplicaciones web 27
2.3.1. Seguridad de la información 28
2.3.2. Requerimientos básicos para la seguridad de la información 29
2.3.3. Seguridad informática 31
2.3.4. Seguridad en las aplicaciones informáticas 31
2.4. Riesgos y vulnerabilidades 33
2.4.1. Riesgos de seguridad en aplicaciones Web 34

2.4.2. Vulnerabilidades en aplicaciones Web 34
2.5. OWASP Top 10 – 2010 37
2.5.1. Inyección 38
2.5.2. Secuencia de Comandos en Sitios Cruzados (XSS) 39
2.5.3. Pérdida de Autenticación y Gestión de Sesiones 40
2.5.4. Referencia Directa Insegura a Objetos 40
2.5.5. Falsificación de Peticiones en Sitios Cruzados (CSRF) 41
2.5.6. Defectuosa Configuración de Seguridad 41
2.5.7. Almacenamiento Criptográfico Inseguro 42
2.5.8. Falla de Restricción de Acceso a URL 43
2.5.9. Protección Insuficiente en la Capa de Transporte 44
2.5.10. Redirecciones y reenvíos no validados 44
CAPÍTULO 3 46
ANALISIS DE LAS APLICACIONES WEB DE LA SBS 46
3.1. Situación Actual de las aplicaciones Web 46
3.2. Sistemas más importantes de la SBS con tecnología JEE 47
3.2.1. Sistema de Población de Identificaciones (SPI) 48
3.2.2. Sistema de Auditoría de Prevención de Lavado de Activos (SAPLA) 48
3.2.3. Sistema Otorgar Credenciales a Intermediarios de Seguros (SOCI) 50
3.2.4. Herramienta de Apoyo al Manual Único de Supervisión (HAMUS) 51
3.3. Arquitectura de las aplicaciones Web de la SBS 53
3.3.1. Enterprise JavaBeans 55
3.3.2. Servicios JEE 56
3.4. Identificación de riesgos en base al Top 10 de OWASP 56
3.5. Factores de Riesgo para la estimación 60
3.5.1. Factores relacionados con el agente causante de la amenaza 61
3.5.2. Factores que afectan a la vulnerabilidad 63
3.6. Factores para estimar el impacto 64
3.6.1. Factores de impacto técnico 64
3.6.2. Factores de impacto sobre el negocio 66
3.6.3. Determinación de la gravedad del riesgo 68
3.6.4. Determinando la severidad 71
3.6.5. Decidir qué se debe corregir 72
3.6.6. Personalización del modelo de calificación de riesgos 73
3.7. Evaluación comparativa de los riesgos identificados 74
CAPÍTULO 4 79

PROPUESTA DE BUENAS PRÁCTICAS DE SEGURIDAD 79
4.1. Buenas prácticas 79
4.2. Buenas prácticas de seguridad para el desarrollo web 81
4.3. Estrategia 84
4.4. Generación de buenas prácticas en el desarrollo de software 86
CAPÍTULO 5 90
CONCLUSIONES Y RECOMENDACIONES 90
5.1. Conclusiones 90
5.2. Recomendaciones 91
BIBLIOGRAFÍA Y WEBGRAFÍA 92

ÍNDICE DE TABLAS

Tabla 1: Análisis de los sistemas en base al top 10 de owasp 58
Tabla 2: Probabilidad de ocurrencia 69
Tabla 3: Impacto de ataque 70
Tabla 4: Probabilidad de ocurrencia 70
Tabla 5: Impacto del ataque..... 71
Tabla 6: Riesgos 75

ÍNDICE DE FIGURAS

Figura 1: Protocolo HTTP 21
Figura 2: Arquitectura Web en 3 capas 26
Figura 3: Seguridad de la información 29
Figura 4: Inyección 39
Figura 5: Secuencias de comandos en sitios cruzados 39
Figura 6: Pérdidas de autenticación y gestión de sesiones 40
Figura 7: Referencia directa insegura a objetos 40
Figura 8: Falsificación de peticiones en sitios cruzados (CSRF) 41
Figura 9: Defectuosa configuración de seguridad 42
Figura 10: Almacenamiento criptográfico inseguro 43
Figura 11: Falla de restricción de acceso a URL 44
Figura 12: Protección insuficiente en la capa de transporte 44
Figura 13: Redirecciones y reenvíos no validados 45
Figura 14: Arquitectura Java Enterprise Edition. 54

RESUMEN

En la actualidad las aplicaciones web se han vuelto indispensables para el manejo de la información en una organización, convirtiéndose en una herramienta que permite al usuario acceder y utilizar un sistema informático a través de internet mediante un navegador web, permitiendo el acceso a la información desde cualquier parte del mundo. La Superintendencia de Bancos y Seguros al ser una institución Pública se ha visto obligada a la adopción de estándares abiertos y software libre para automatizar sus procesos, y ha desarrollado aplicaciones web utilizando la plataforma Java Enterprise Edition (JEE) sin embargo no se ha aplicado ningún tipo de estándar o buenas prácticas en el aseguramiento del aplicativo. El presente proyecto tiene como objetivo el análisis de riesgos de las aplicaciones web utilizando las recomendaciones OWASP Top 10 – 2010 para descubrir las vulnerabilidades que se presenta durante el desarrollo de un software y estimar el riesgo asociado para el negocio. A partir de los resultados obtenidos donde se identificaron la ocurrencia de almacenamiento criptográfico inseguro y protección insuficiente en la capa de transporte se realizó una propuesta de buenas prácticas para asegurar las aplicaciones, corregir los riesgos detectados y asegurar el proceso de desarrollo de nuevas funcionalidades y existentes.

Palabras Claves: OWASP, Seguridad, Riesgo, Desarrollo y Aplicaciones Web.

ABSTRACT

Nowadays Web applications have become essential to the management of information in an organization, making it a tool that allows users to access and use a computer system via the Internet using a web browser, allowing access to information from anywhere in the world. The Superintendencia de Bancos y Seguro is a public institution has been forced to adopt open standards and free software to automate their processes, and developed web applications using Java Platform, Enterprise Edition (JEE) but has not been applied any standard or good practices in ensuring the application. This project aims to risk analysis of web applications using the recommendations OWASP Top 10 - 2010 to discover vulnerabilities that occurs during software development and estimate the associated risk to the business. From the results obtained where the occurrence of insecure cryptographic storage and insufficient protection in the transport layer of a proposal identified good practices to ensure applications made, correct the identified risks and ensure the process of developing new features and existing.

KeyWords: OWASP, Security, Risk, Development and Web Applications.

GLOSARIO DE NOMENCLATURAS

- SBS: Superintendencia de Bancos y Seguros.
- CGT: Coordinación General de Tecnología.
- SDAT: Subdirección de Desarrollo y Aplicaciones Tecnológicas.
- OWASP: The Open Web Application Security Proyect.
- R: Riesgo.
- HTTP: Protocolo de transferencia de hipertexto.
- URL: Uniform Resource Locator.
- JEE: Java Enterprise Edition.
- CAS: Central Authentication Service.
- SOCI: Sistema para Otorgar Credenciales a Intermediarios de Seguros.
- SAPLA: Sistema de Auditorias para la Prevención de Lavado de Activos.
- SPI: Sistema de Población de Identificaciones.
- LDAP: Lightweight Directory Access Protocol.
- AES: Advance Encryption Standard.
- SSL: Secure Sockets Layer.
- TLS: Transport Layer Security.
- MD5: Messege Digest Algorithm 5.

CAPÍTULO 1

ANÁLISIS DE LAS APLICACIONES WEB DE LA SUPERINTENDENCIA DE BANCOS Y SEGUROS, UTILIZANDO LAS RECOMENDACIONES TOP TEN DE OWASP PARA DETERMINAR LOS RIESGOS MÁS CRÍTICOS DE SEGURIDAD E IMPLEMENTAR BUENAS PRÁCTICAS DE SEGURIDAD PARA EL DESARROLLO DE SUS APLICATIVOS.

En la actualidad las aplicaciones web se han vuelto indispensables para el manejo de la información en una organización, convirtiéndose en una herramienta que permite al usuario acceder y utilizar un sistema informático a través de internet mediante un navegador, permitiendo el acceso a la información desde cualquier parte del mundo sin importar el sistema operativo que se utilice.

Se ha comprobado que en los últimos años, 75% o más de los ataques electrónicos fueron a nivel de aplicación (y no a nivel de host o de red).

Todo tipo de transacciones se realizan actualmente en la web, y cada vez en mayor proporción. Detalles de cuentas bancarias, tarjetas de crédito y todo tipo de información confidencial y de valor circulan en enormes cantidades y continuamente. Por lo tanto es lógico que la mayoría de los esfuerzos de hackers y demás atacantes se centre en vulnerar estas aplicaciones.

La variedad y complejidad de los requerimientos de usuarios finales continúa creciendo, y con ellos aumenta la complejidad de las aplicaciones, la cantidad de funcionalidades y fases de pruebas. Esto sumado al incremento en la competencia y en la necesidad de superarla en el time-to-market, implican sacrificios importantes en los aspectos no-funcionales de la aplicación y

específicamente en los aspectos de seguridad. Especialmente cuando no existe una conciencia de seguridad a nivel corporativo, se tiende a generar una alta presión para terminar el trabajo sin considerar suficientemente las posibles vulnerabilidades o se emplean herramientas y tecnologías sin tener en cuenta tan importante aspecto (Scambray & Shema, 2002)

Las aplicaciones web están en parte definidas por su uso del protocolo HTTP como medio de comunicación entre cliente y servidor. Este protocolo es simple y basado en ASCII, no se requiere gran esfuerzo para generar pedidos y descifrar el contenido de las respuestas; además utiliza un puerto TCP bien conocido, de poco sirve un firewall para proteger una aplicación si tiene que admitir el tráfico a través del puerto 80.

Este protocolo no mantiene por sí mismo el estado de la sesión un atacante no tiene que emular mecanismos de mantenimiento de sesión, basta con emitir una solicitud para lograr el cometido. Mecanismos como el uso de cookies permiten simular una sesión virtual intercambiando información adicional en cada request/response, pero no son efectivas si no se las implementa bien, e introducen problemas adicionales de seguridad y privacidad.

Existen muchas excepciones y variantes adicionales a estos elementos; en particular se utiliza ampliamente SSL como protocolo de encriptación a nivel de transporte en las comunicaciones cliente-servidor (Microsoft, 2003).

Algunos de los mitos más comunes en las empresas de software son los que relaciona a continuación y sobre ellos muchas empresas descansan la seguridad de sus aplicaciones: (Naidu, 2003)

- El usuario solamente enviará entradas esperadas : HTML admite el uso de etiquetas (tags) que manipulan la entradas a la aplicación, por ejemplo si la aplicación utiliza campos ocultos para enviar información sensible estos pueden ser fácilmente manipulados desde el cliente.
- Otro mito común es que la validación se puede realizarse únicamente del lado del cliente con JavaScript: si no se efectúa ninguna validación del lado del servidor, cualquier atacante que evite esta validación (para nada difícil de lograr) tendrá acceso total a toda la aplicación.
- Comúnmente se cree que el uso de firewalls es suficiente: si el firewall tiene que habilitar los puertos 80 y/o 443 para que la aplicación sea accesible al exterior, no podrá hacer nada para detectar entradas maliciosas del cliente, y por supuesto no es protección contra ataques internos.
- El uso de SSL es una solución suficiente: SSL simplemente cubre el request/response HTTP dificultando la intercepción del tráfico entre cliente y servidor, pero no agrega seguridad al servidor ni evita el envío de código malicioso desde el cliente.

Aunque solo se han analizado algunos de los múltiples aspectos relativos a la seguridad en las aplicaciones web es suficiente para comprobar lo fácilmente que puede ser vulnerada una aplicación cuando no se le asigna una prioridad adecuada a los controles de seguridad en las distintas etapas de desarrollo.

La presente realidad de la industria de software atenta contra la posibilidad de implementar estos controles en forma adecuada, en particular la creciente

complejidad y variedad de tecnologías incrementa de la misma forma la variedad de puntos vulnerables y técnicas de ataque.

Muchas de las vulnerabilidades que se pueden presentar son propias de la plataforma sobre la que se desarrolla la aplicación (Sistema Operativo, software de base, herramientas de desarrollo), otras son negligencia por parte de jefes de proyecto, arquitectos, diseñadores, programadores, administradores y usuarios del sistema.

Se ha apreciado que varias medidas de control, deben ser implementadas en el marco de políticas de seguridad establecidas, ejecutadas en varias fases distintas del ciclo de vida de la aplicación, y controladas por un auditor, que permitan disminuir considerablemente los riesgos e impacto de estas amenazas vistas, aunque difícilmente sea posible asegurar la invulnerabilidad de una aplicación.

1.1. Antecedentes

Con el auge del uso de aplicaciones en la web se ha incrementado también las tecnologías para brindar cada vez soluciones más profesionales y lograr mayor rapidez en el desarrollo. Tal es el caso de lenguajes como Php, Java, C#, Python, frameworks y tecnologías como son J2EE, Symfony, Django, etc.

J2EE es el acrónimo de Java 2 Enterprise Edition diseñado por Sun Microsystems y define un estándar para el desarrollo de aplicaciones empresariales multicapas.

Simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proporcionando un completo conjunto de servicios a través de estos componentes, y manejando muchas de las funciones de la

aplicación de forma automática, sin necesidad de una programación compleja (Gosling, 2002). En otras palabras, la idea subyacente a la plataforma J2EE es proporcionar un estándar sencillo y unificado para aplicaciones distribuidas en un modelo de aplicación basado en componentes. Está basado en J2SE (Java 2 Standard Edition) manteniendo muchas de sus características, como su portabilidad “write once, run anywhere” (Jonhson, 2003), pero añade una serie de elementos necesarios en entornos empresariales, relativos a redes, acceso a datos y entrada/salida que requieren mayor capacidad de procesamiento, almacenamiento y memoria. La decisión de separarlos es debido a que no todas estas características son necesarias para el desarrollo de aplicaciones estándar.

Las aplicaciones J2EE están compuestas de diferentes componentes. Un componente J2EE es una unidad de software funcional auto-contenido que se ensambla dentro de una aplicación J2EE con sus clases de ayuda y ficheros y que se comunica con otros componentes de la aplicación. La especificación J2EE define los siguientes componentes J2EE:

- Las Aplicaciones Clientes y los Applets son componentes que se ejecutan en el lado del cliente.
- Los componentes Java Servlet la tecnología JavaServer Pages son componentes Web que se ejecutan en el lado del servidor.
- Los Enterprise JavaBeans (beans enterprise) son componentes de negocio que se ejecutan en el servidor de aplicación.

J2EE ofrece muy buenas perspectivas para la implementación de software

empresarial. Entre las ventajas que ofrece se pueden citar las siguientes:

- **Soporte para múltiples sistemas operativos:** es posible desarrollar arquitecturas basadas en J2EE usando cualquier sistema operativo donde pueda ejecutarse una máquina virtual de Java, teniendo la gran ventaja de una independencia total de la arquitectura de hardware.
- **Objetos gestionados:** proporciona un entorno gestionado para componentes y las aplicaciones son céntricas respecto al contenedor. Al ser gestionados, los componentes utilizan la infraestructura proporcionada por los servidores de J2EE sin que el programador sea consciente de ello. Las aplicaciones de J2EE son también declarativas, un mecanismo con el que puede modificar y controlar el funcionamiento de las aplicaciones sin cambiar de código (Beust, 2002)
- **Reusabilidad:** la separación de los requisitos de una aplicación en sus partes integrantes es un modo de conseguir la reutilización; utilizar la orientación a objeto para encapsular la funcionalidad compartida es otro. Sin embargo, a diferencia de los objetos, los componentes distribuidos requieren una infraestructura más compleja para su construcción y gestión. J2EE ofrece una arquitectura notablemente rigurosa para la construcción y gestión de componentes distribuidos así como su reutilización de componentes. Además, ya que los componentes son reproducibles, lo cual quiere decir que es posible identificar ciertos metadatos sobre los componentes, las aplicaciones pueden ser creadas

componiendo tales componentes. Ambas características fomentan la reutilización del código a alta granulometría (Beust, 2002).

- **Modularidad:** es siempre recomendable dividir una aplicación en módulos discretos, cada uno de ellos responsable de una tarea específica, logrando que las aplicaciones sean mucho más fáciles de mantener y comprender. Por ejemplo, los servlets de Java, las JSP (Java Server Page) y EJB (Enterprise Java Beans) proporcionan un modo de modularizar las aplicaciones, fragmentando las aplicaciones en diferentes niveles y tareas individuales (Beust, 2002).
- **Fácil integración con los sistemas de información existentes:** JDBC (Java Database Connectivity), una tecnología J2EE, es una API de Java para bases de datos SQL, lo que permite que se acceda a cualquier tipo de información recogida en tablas que pueda existir. JNDI (Java Naming and Directory Interface) permite a las aplicaciones que utilizan tecnología Java tener acceso a los servicios de nomenclatura y directorio de la empresa.
- **Gran variedad de herramientas, servidores y componentes:** para desarrollar las aplicaciones que necesiten el equipo de desarrollo puede seleccionar las soluciones que mejor se adapten a sus necesidades, sin tener que ajustarse a las ofertas de un solo fabricante.
- **Organismo de control:** J2EE está controlada por un organismo formado por más de 400 empresas. Entre esas empresas se encuentran

muchas de las más importantes del mundo informático, tales como Sun Microsystems, IBM, Oracle, BEA, HP, AOL, etc.

- **Competitividad:** muchas empresas crean soluciones basadas en J2EE que ofrecen características tales como rendimiento y precios muy diferentes. De este modo, se ha desarrollado a un nivel exponencial la plataforma y los clientes tienen la posibilidad de escoger entre una gran cantidad de opciones.
- **Madurez:** creada en el año 1997, J2EE ya tiene varios años de vida y una amplia cantidad de proyectos importantes están desarrollados sobre esta plataforma.
- **Soluciones libres:** sobre la plataforma J2EE es posible crear arquitecturas basadas por completo en productos de software libre. No solo eso, sino que los arquitectos de software disponen de muchas soluciones libres para cada una de las partes de su arquitectura.

La plataforma J2EE posee un modelo de desarrollo del cual es importante tener conocimiento de algunos conceptos básicos sobre el modelo de desarrollo de aplicaciones bajo esta plataforma. La plataforma J2EE define un modelo de programación encaminado a la creación de aplicaciones basadas en n-capas. La lógica de la aplicación se divide en componentes con diferentes funciones y que están distribuidos en un ambiente multicapas. Aunque puede variar, típicamente una aplicación suele tener cinco capas diferentes:

- Capa cliente: representa la interfaz de usuario que interactúa con el cliente.

- Capa de presentación: representa el conjunto de componentes que generan la información que se mostrará en la interfaz de usuario del cliente. Normalmente se crea a través de componentes basados en Servlets y JSP.
- Capa de lógica de negocio: contiene los componentes de negocio reutilizables.
- Capa de integración: aquí se encuentran los componentes que permiten hacer más transparente el acceso a la capa de sistemas de información. Este es el lugar idóneo para implementar la lógica de objetos de acceso a datos (DAO, data access object).
- Capa de recursos: engloba los sistemas en los cuales la información se almacena físicamente como bases de datos relacionales, sistemas legacy (heredados), bases de datos orientadas a objetos, bancos de ficheros de datos, etc.

Las ventajas de un modelo como este son muy importantes. Al tener las capas separadas existe un bajo acoplamiento entre las mismas, de modo que es mucho más simple hacer modificaciones en ellas sin que afecten a las demás. Todo esto conlleva a la obtención de mejoras en cuanto a mantenibilidad, extensibilidad y reutilización de componentes. Otra ventaja que se obtiene es la de promover la heterogeneidad de los clientes, ya que añadir nuevos tipos de clientes se reduce a añadir nuevas capas de interfaz de usuario y presentación, sin necesidad de modificar el resto de las capas. Como ya se ha dicho, el modelo de desarrollo con J2EE está basado en componentes

reutilizables, con el objetivo de aumentar la reusabilidad de las aplicaciones. Estos componentes, además, gracias a las especificaciones, son intercambiables entre servidores de aplicaciones, por lo que la portabilidad de las aplicaciones es máxima.

En el mundo del desarrollo web, el tema de seguridad es un tema crucial. Existen diversas vulnerabilidades que son focos de riesgo y puerta de entrada a atacantes de todo tipo, que pueden afectar el correcto funcionamiento de las aplicaciones o la integridad y privacidad de datos sensibles. (Web Seguro, 2012)

La Open Web Application Security Proyect (OWASP), es un proyecto de código abierto de seguridad en aplicaciones web, especializado en determinar y combinar las causas que hacen un software inseguro, provee asistencia para mejorar la seguridad de las aplicaciones difundiendo información sobre las vulnerabilidades y fallos en sus guías para que las organizaciones y desarrolladores puedan aplicar y evitar riesgos reales de seguridad.

OWASP ofrece un Top 10 sobre los diez riesgos más críticos sobre la seguridad en aplicaciones, cada ítem describe la probabilidad general y los factores de consecuencia que se utiliza para clasificar la gravedad típica del riesgo y finalmente orienta como verificar si posee problemas en esta área, como evitarlos, ejemplos e información, con el objetivo principal de educar a desarrolladores, diseñadores, arquitectos, gerentes, y organizaciones sobre las consecuencias de las vulnerabilidades de seguridad más importantes en aplicaciones web.

Los riesgos que cubre el Top Ten de OWASP son los siguientes:

1. Inyección.
2. Secuencia de Comandos en Sitios Cruzados (XSS).
3. Pérdida de Autenticación y Gestión de Sesiones.
4. Referencia Directa Insegura a Objetos.
5. Falsificación de Peticiones en Sitios Cruzados (CSRF).
6. Defectuosa Configuración de Seguridad.
7. Almacenamiento Criptográfico Inseguro.
8. Falla de Restricción de Acceso a URL.
9. Protección Insuficiente en la Capa de Transporte
10. Redirecciones y reenvíos no validados

El análisis de las aplicaciones web de la Superintendencia de Bancos Seguros del Ecuador mediante el top ten de OWASP permitirá estimar el estado real de la seguridad de un sistema, clasificar los riegos encontrados (alto, medio, bajo) y establecer buenas prácticas de seguridad en el desarrollo de las aplicaciones web.

La Superintendencia de Bancos y Seguros del Ecuador (SBS), es un organismo del sector Público, cuya finalidad esencial es la supervisión de las operaciones bancarias a nivel nacional, es decir, velar por la seguridad, estabilidad, transparencia y solidez de los sistemas financieros para proteger los intereses del público e impulsar el desarrollo del país. Sus sistemas y centros de información necesariamente deben asegurar la calidad y la seguridad de la información, y el servicio informático, con tecnología de punta.

La SBS, al ser un organismo Público se ha visto obligada a la adopción de estándares abiertos y software libre para el desarrollo de sus aplicaciones, así

como la creación de sitios Web institucionales para automatizar los procesos internos y externos, las mismas que tendrán una arquitectura N capas y orientadas a la Web, para ello la SBS cuenta con el apoyo de la Coordinación General de Tecnología (CTG) la que dispone de una área especializada para el desarrollo de software, denominada Subdirección de Desarrollo y Aplicaciones Tecnológicas (SDAT).

La Superintendencia de Bancos y Seguros ha desarrollado aplicaciones empresariales en varias plataformas, desde hace varios años. Para la implementación de las aplicaciones más importantes de la institución se ha utilizado la plataforma Java Enterprise Edition, sin embargo no se ha aplicado ningún tipo de estándar en el aseguramiento del aplicativo.

Actualmente la SBS sugiere los siguientes estándares de seguridad para el desarrollo de sus nuevas aplicaciones Web.

- El sistema deberá asegurar confidencialidad sobre los datos del usuario. Cada usuario tiene un nombre único y una clave segura, mayor a 6 caracteres y con combinación de letras y números, que serán almacenados de manera cifrada.
- Todos los datos con información sensible del usuario deberán viajar encriptados mediante la utilización de SSL.
- Siempre se manejaran dos tipos de usuarios principales: los usuarios registrados y los funcionarios internos. Cada perfil dispondrá para realizar las operaciones pertinentes y no podrán acceder a privilegios de usuarios que posean otro perfil.

- Se debe contar con un mecanismo de Single-Sign-On para la autenticación con una clave única en todos los sistemas.

Es de prioridad de la SBS analizar los riesgos y vulnerabilidades más significativas de sus aplicaciones Web, las mismas que permitirán establecer buenas prácticas y un punto de referencia para empezar a implementar los estándares de seguridad y programar de una manera orientada a la seguridad.

1.2. Justificación

La SBS se beneficiará de este proyecto, porque en los resultados del análisis de las aplicaciones web se encontrarán los riesgos y vulnerabilidades más significativos haciendo uso del Top 10 de OWASP, donde se podrán hacer rectificaciones a futuro en las políticas de seguridad establecidas, así como, educar a los desarrolladores, subdirectores y coordinadores de la CTG en la seguridad de las aplicaciones web y cómo tenerla en cuenta al desarrollarlas.

1.3. Objetivos

1.3.1. Objetivo General

Analizar las aplicaciones web de la SBS para identificar los riesgos de seguridad más comunes mediante el top ten de OWASP y definir buenas prácticas para el aseguramiento de las aplicaciones.

1.3.2. Objetivo Específicos

- Identificar las aplicaciones web más importantes de la Superintendencia de Bancos y Seguros bajo la plataforma JEE.
- Identificar las vulnerabilidades, amenazas y riesgos más comunes presentes en una aplicación Web.
- Realizar una evaluación de riesgos basados en el Top Ten de OWASP.

- Recomendar la implementación de buenas prácticas en base a los riesgos identificados.

1.4. Alcance o Meta

El presente proyecto cubre el análisis de las aplicaciones web más importantes de la institución bajo la plataforma Java Enterprise Edition (JEE) y que cumplen los estándares de seguridad actuales de la SBS, contempla su análisis en base al Top 10 de OWASP y la identificación de los mayores riesgos para empezar un programa de buenas prácticas en el desarrollo de aplicaciones; no contempla el plan de seguridad informática.

Los resultados de este proyecto servirán para que los desarrolladores de la Subdirección de Desarrollo y Aplicaciones Tecnológicas, puedan identificar los riesgos a tomar al momento del desarrollo de los aplicativos con la finalidad de mantener la información confiable y segura.

1.5. Metodología de Investigación

La metodología seleccionada dentro de este proyecto de tesis está basada en la Investigación Contrastiva e investigación Aplicativa.

La investigación contrastiva, cubre la necesidad de encontrar los errores de las teorías del mismo, con el objetivo de eliminarlas, reajustarlas o aumentar su veracidad.

La investigación aplicativa, parte del hecho de que, dentro de la secuencia de trabajo, existen teorías cuya veracidad se ha elevado gracias a un cierto número de aplicaciones y además, del hecho de que en el mundo de las necesidades de desarrollo existen requerimientos que pueden ser satisfechos aprovechando esas teorías. Su objetivo central está en proveer tecnologías o

esquemas de acción derivados de los conocimientos teóricos construidos dentro de la secuencia de la línea. Esta investigación tiende a establecer una relación productiva, ingeniosa y creativa, entre las posibilidades de un modelo teórico, por un lado, y las necesidades que se confrontan en el terreno de la práctica.

En resumen, la investigación contrastiva se utilizará en la primera etapa del proyecto de tesis, luego de un análisis identificar los riesgos y vulnerabilidades encontradas en el proceso de desarrollo de aplicaciones web de la SBS, en base al Top 10 de OWASP. En la segunda parte del proyecto se utilizará la investigación aplicativa ya que el fin del proyecto de tesis es empezar a establecer un programa de buenas prácticas en las aplicaciones web de acuerdo al Top 10 de OWASP al proceso de desarrollo de sus aplicativos.

Se pueden destacar los Métodos Científicos utilizados en la investigación:

- **Métodos Lógicos:** El método analítico-sintético al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos de forma independiente, para luego sintetizarlos en la solución de la propuesta.
- **Métodos Empíricos:** El método análisis de las aplicaciones web de la SBS para identificar los riesgos de seguridad más comunes mediante el top ten de OWASP y principales dificultades existentes para el aseguramiento de las aplicaciones.
- **Método experimental:** Para evaluar resultados a partir de la experimentación y valoración de las muestras.

- **Método Hipotético-deductivo:** Para la elaboración de la hipótesis central de la investigación y desarrollar procedimientos que arriben a conclusiones particulares.

CAPÍTULO 2

MARCO TEÓRICO

El reciente desarrollo de la tecnología nos ha hecho posible vivir en algunos aspectos, mejor de lo que se podia pensar en el pasado. Hoy se puede acceder a masivas cantidades de información en Internet, se puede experimentar con algún juego online a través de internet, es decir, que la tecnología informática a continuando mejorando nuestra calidad de vida tanto a nivel laboral como personal.

Las aplicaciones informáticas se han vuelto una parte importante en nuestras vidas, incluso para los que no utilizan ordenadores, ya que muchos de sus datos están informatizados. Algunas de estas mejoras han sido muy sutiles, y han pasado desapercibidas y quizá no se le dio la importancia que debería. Hasta hace bien poco, las cosas se hacían de otra manera, y la informática ha cambiado algunos de nuestros hábitos.

La mayoría de los sistemas tradicionales con los que se interactúa hoy, pueden caer hasta cierto punto en la clasificación de aplicaciones de escritorio y aplicaciones web. Las aplicaciones de escritorio necesitan estar instaladas en el ordenador del usuario que la va a utilizar, lo cual a veces representa un problema si no se tienen los permisos adecuados para operarla o instalarla. Las aplicaciones web, con el surgimiento de internet y la generalización del uso de los móviles se han vuelto muy populares y representan hoy la gran mayoría de las aplicaciones, pues tienen la ventaja de que se puede acceder a ella desde cualquier lugar en el cual se tenga una conexión a internet, sin importar

la plataforma en la cual se encuentre uno trabajando en esos momentos y varios usuarios pueden estar operando en la misma aplicación a la vez.

2.1. Introducción a las aplicaciones web y la seguridad

Las aplicaciones web deben su éxito espectacular a dos pilares fundamentales: el protocolo HTTP y el lenguaje HTML. El primero permite una implementación simple y sencilla de un sistema de comunicaciones que facilita el envío de cualquier tipo de ficheros de una forma fácil, simplificando el funcionamiento del servidor y permitiendo que servidores poco potentes atiendan miles de peticiones y reduzcan los costes de despliegue. El otro nos proporciona un mecanismo de composición de páginas enlazadas simple y fácil, altamente eficiente y de uso muy simple.

En sus inicios la web era simplemente una colección de páginas estáticas, documentos, etc., que podían consultarse o descargarse. Luego fue evolucionando a métodos para confeccionar páginas dinámicas que permitiesen que lo mostrado fuese dinámico.

Las aplicaciones web presentan varias ventajas entre como la compatibilidad multiplataforma, el acceso de múltiples usuarios de forma concurrente, la información en línea, facilidad de actualización, entre otras.

Al estar las aplicaciones expuestas al acceso a través de internet, se ciernen sobre ellas diversas amenazas de orígenes diversos; es por eso que hablar de aplicaciones web se encuentra ligado permanentemente al tema seguridad.

Los ataques a nivel de aplicación son una amenaza en constante aumento contra la seguridad Web. Para esto se utilizan tradicionalmente una gran

variedad de medios para paralizar un sitio Web e introducirse en él, lo que provoca resultados que varían desde un menor rendimiento del sitio Web hasta robos de datos y la desprotección de la infraestructura.

2.1.1 Cliente – Servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras (Mateu, Desarrollo de Aplicaciones Web, 2004).

- **Cliente:** Comúnmente es una computadora o una aplicación informática que de forma remota consume un servicio hacia un servidor que a su vez es otra computadora que son parte de una misma red.
- **Servidor:** Proporciona servicios a estaciones o nodos llamados clientes, es decir, el servidor es un programa que recibe múltiples solicitudes y devuelve una respuesta con resultados.

El protocolo mediante el que se comunican el cliente y el servidor es conocido como HTTP. El cual se encuentra estandarizado y no ha de ser creado por el programador de aplicaciones.

El protocolo HTTP forma parte de la familia de protocolos de comunicaciones TCP/IP, que son los que se utilizan en internet. Estos

protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre distintos ordenadores.

2.1.2 Protocolo HTTP y HTTPS

Un protocolo es un conjunto de reglas usadas para el intercambio de mensajes a través de una red, estos protocolos viajan en grupos denominados paquetes. Los protocolos se pueden dividir en varias categorías, la más estudiada es la OSI (Open System Interconnection) (Quero Catalinas, García Román, & Peña Rodríguez, 2007). Este modelo creado por la Organización Internacional para la Estandarización (ISO) en el año 1980 (ZIMMERMAN, 1980), es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones.

El modelo OSI está dividido en siete capas (Pérez Herrera, 2003), las cuales son la Capa de Aplicación, en la cual se encuentran los servicios de red a aplicaciones FTP, HTTP, HTTPS, SMTP, SSH, la Capa de Presentación, la Capa de Sesión, la Capa de Transporte, la Capa de Enlace de Datos, la Capa de Red y la Capa Física.

Se describe solamente los protocolos HTTP y HTTPS, pues es sobre ellos es que se accede a las aplicaciones a las que se hará referencia en la presente tesis.

El protocolo de transferencia de hipertexto, conocido como HTTP, actúa como un protocolo cliente – servidor mediante el cual se transfiere información entre los clientes Web y los servidores HTTP, es el más utilizado en internet.

HTTP se basa en sencillas operaciones de solicitud/respuesta como se indica en la Figura 1. Un cliente establece una conexión con un servidor y envía

un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan identificado mediante un localizador uniforme de recursos (URL) (Donate, 2005). Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

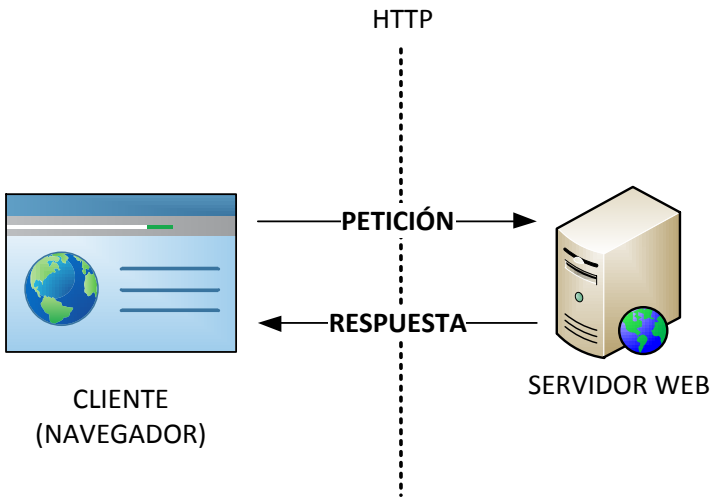

Figura 1: Protocolo HTTP

El protocolo seguro de transferencia de hipertexto, conocido como HTTPS, cifra los datos al momento de transmitirlos de una forma más segura a través de internet, está basado en el protocolo HTTP. HTTPS es utilizado generalmente en instituciones financieras por solicitar datos personales y contraseñas o cualquier sitio web que necesite información personal para completar una transacción en línea.

El sistema HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador

utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar. El puerto estándar de para este puerto es el 433 (Mateu, 2004). Para el protocolo HTTP se utiliza generalmente el puerto 80.

2.2. Aplicaciones Web

Una aplicación web es un tipo especial de aplicación cliente/servidor la cual puedes ser utilizada por los usuarios accediendo a un servidor web utilizando un navegador a través de internet o intranet.

Surgieron en los 90 primeramente potenciados por el surgimiento de las redes de computadoras y la necesidad de compartir contenidos y ficheros a través de estas.

Este tipo de aplicación se ha vuelto muy popular por la facilidad de actualización y mantenimiento de las aplicaciones sin la necesidad de instalar o distribuir software en los clientes. A la interfaz de usuario de una aplicación web se accede normalmente desde un navegador web, los cuales hoy día no dependen de un sistema operativo específico para su ejecución. Las aplicaciones más utilizadas en la actualidad o las más populares para acceder a las aplicaciones web son: Firefox, Chrome, Internet Explorer, Opera y Safari.

2.2.1. Evolución de las Aplicaciones web

La programación web es un término adecuado para describir el proceso general que engloba el diseño y la creación de un sitio web. En los años 90, los

sitios web eran utilizados solamente como folletos digitales donde el principal uso era el de mostrar información. Actualmente los sitios son más grandes y complejos.

Con el desarrollo de las tecnologías, de internet, la aparición de las aplicaciones de comercio electrónico y las páginas dinámicas, los sitios ya han dejado de ser solamente folletos digitales y han pasado a ser auténticas aplicaciones de software.

La Web 1.0, surgida en la década de los 90, es la forma más básica que existe, con navegadores que solamente procesan texto, lo cual los hace ser bastante rápidos pues solo ofrecen la lectura de información. El usuario, no puede interactuar con el contenido de la página, estando totalmente limitado a lo que los administradores publiquen en ésta (Fuchs & et al, 2010).

Algunos elementos que nos permiten identificar los diseños típicos de un sitio de la conocida como Web 1.0 son:

- Páginas estáticas en vez de dinámicas por el usuario que la visita.
- El uso de framesets o Marcos.
- Libros de visitas online o guestbooks.
- Formularios HTML enviados vía email.
- No se pueden introducir comentarios ni otro tipo de contenido.
- Todas sus páginas se crean de forma estática y muy pocas veces se actualizan.
- Cuando una página es actualizada no se trata de una nueva versión, sino de una nueva forma de ver y organizar las cosas.

El término Web 2.0 abarca aquellos sitios web que facilitan y se basan en el hecho de compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración web. (Areitio, 2008).

Un sitio que cae dentro de la denominación de Web 2.0, permite a los usuarios interactuar y colaborar entre sí como creadores de contenido, el cual es generado por usuarios en una comunidad virtual, a diferencia de sitios web estáticos donde los usuarios se limitan a una observación pasiva de los contenidos que se han creado para ellos.

Se encuentra varios ejemplos de la Web 2.0, entre los cuales se hallan las comunidades, los servicios web, las aplicaciones web, las redes sociales, los servicios de alojamiento de videos, las wikis, los blogs, etc. (O'Reilly, 2007), como pudieran ser Applesfera, Wikipedia, Twitter, Facebook, Youtube, Dropbox, entre otros.

La Web 3.0 es una clasificación que se utiliza para describir la evolución del uso y la interacción de las personas en internet a través de diferentes formas entre los que se incluyen la transformación de la red en una gran base de datos, donde se crean contenidos accesibles por múltiples aplicaciones, se aplican tecnologías de inteligencia artificial, la web semántica, la web 3D, etc. Se basa fundamentalmente en la idea de añadir metadatos semánticos y ontológicos a la World Wide Web. Esas informaciones adicionales que describen el contenido, el significado y la relación de los datos se deben proporcionar de manera formal, para que así sea posible evaluarlas automáticamente por máquinas de procesamiento.

El objetivo es mejorar Internet, ampliando la interoperabilidad entre los sistemas informáticos usando "agentes inteligentes". Los agentes inteligentes son programas que buscan información sin operadores humanos. Con la web 3.0 se busca que los usuarios puedan conectarse desde cualquier lugar, cualquier dispositivo y en cualquier momento (Hendler, 2009).

Entre sus innovaciones destacan Bases de datos, Inteligencia artificial, Web semántica y SOA, Evolución al 3D.

En la web 4.0, el cual es un novedoso concepto en la actualidad, las aplicaciones ya no estarán en las PC’s, estarán en la internet y en principio accesibles desde cualquier lugar. Se utilizara una red inteligente donde el objetivo primordial será el de unir las inteligencias donde tanto las personas como las cosas se comuniquen entre sí para generar la toma de decisiones. En el futuro se espera que haya “agentes” en la Web que conozcan, aprendan y razonen como lo hacen las personas (Aghaei, Nematbakhsh, & Khosravi Farsani, 2012).

Ya hoy se está hablando de un nuevo concepto que está aún en desarrollo y es conocido como la Web Ubicua (López Trujillo, Marulanda Echeverry, & Vega, 2012); la idea que se persigue es que se vayan complementando algunas tecnologías y lo demás dependerá de la imaginación de los desarrolladores y de las principales empresas del sector, pues es elevadísima la inversión que se está dedicando a I+D en esta dirección, donde ya se observan tecnologías futuristas como Google Glass, Microsoft Surface, Siri de Apple, etc. Y que sin duda alguna está cambiando la manera de explotar la información empresarial y su inversión económica.

2.2.2. Arquitectura en capas.

Aunque una aplicación web puede ser desarrollada de diversas formas, existen muchas variaciones posibles; las aplicaciones web son aplicaciones que generalmente se desarrollan distribuidas en 3 capas o niveles (Figura 2), se utiliza el navegador web como la primera capa, la lógica de negocio sería la capa intermedia, la cual puede desarrollarse en diferentes lenguajes de programación como PHP, Java, Python, Perl, ASP.NET, etc y la base de datos sería la última capa. El navegador web, al cual toma el nombre de cliente envía peticiones a la capa media, la cual las procesa y accede a la base de datos para realizar las transacciones solicitadas. Las peticiones realizadas entre el cliente y el servidor se realizan a través del protocolo HTTP y se emplea el lenguaje HTML para mostrar el contenido de respuesta a la petición inicial entre cliente y servidor.

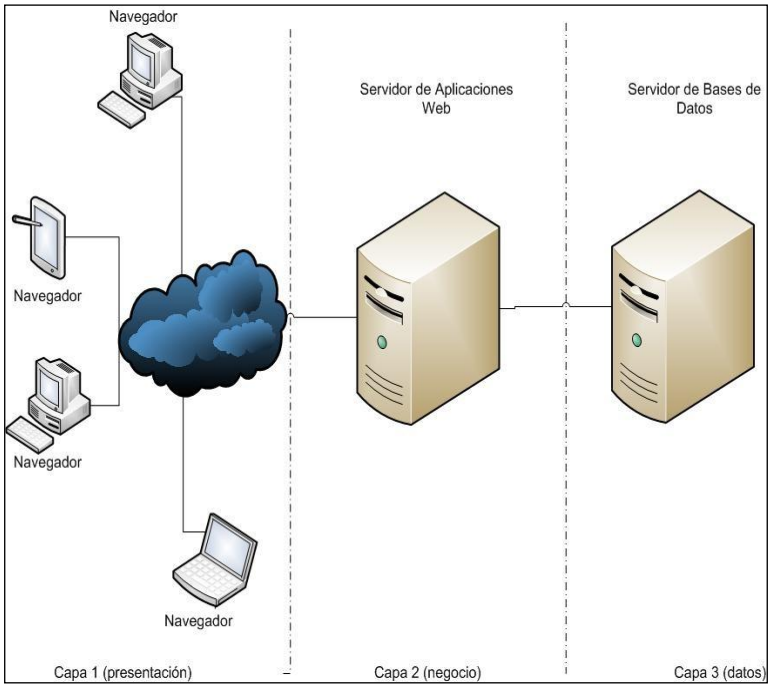

Figura 2: Arquitectura Web en 3 Capas

2.3. Seguridad en las aplicaciones web

Cualquier empresa u organización hoy día tiene que realizar un esfuerzo significativo y una elevada inversión para asegurar que la información y recursos se encuentren protegidos siempre que se encuentren expuestos sus servicios a la red de redes. Internet es un factor primordial en las comunicaciones y también un evidente riesgo potencial de acceso y mal uso de los servicios e información disponibles. Se clasifican como sistemas más críticos aquellos donde la seguridad debe de ser muy significativa, pero en general todas las aplicaciones web deben de estar protegidos y asegurados ante los principales ataques o al menos los más comunes.

En una aplicación web, la seguridad se divide en:

- **Disponibilidad:** Propiedad o característica de los activos consistentes en que las entidades o procesos autorizados tienen acceso a los mismos cuando lo requieren.
- **Autenticidad:** Propiedad o característica consistente en que una entidad es quien dice ser o bien que garantiza la fuente de la que proceden los datos
- **Integridad:** Propiedad o característica consistente en que el activo de información no ha sido alterado de manera no autorizada.
- **Confidencialidad:** Propiedad o característica consistente en que la información ni se pone a disposición, ni se revela a individuos, entidades o procesos no autorizados.

- **Trazabilidad:** Propiedad o característica consistente en que las actuaciones de una entidad pueden ser imputadas exclusivamente a dicha entidad.

Las aplicaciones web son consideradas como el punto más común para los ataques informáticos debido a su fácil acceso a través de internet, muchas de ellas contienen información sensible de instituciones que mueven todo su negocio mediante una aplicación web. Una institución u organización mientras más va automatizando sus procesos mediante aplicaciones web, se vuelve más importante la necesidad de implementar seguridad en sus procedimientos e información (UNAM-CERT, 2009).

La mayoría de los problemas de seguridad en los sitios web se encuentran a nivel aplicación y son el resultado de la escritura de código con problemas. Programar aplicaciones web seguras no es una tarea fácil, pues se requiere por parte del programador, una concepción general de los riesgos que puede correr la información contenida, solicitada y recibida por el sistema. En la actualidad, aunque existen muchas publicaciones que permiten formar un criterio sobre el tema, no existen acuerdos básicos sobre lo que se debe o no se debe hacer (Areitio, 2008).

2.3.1. Seguridad de la información

No se debe confundir la seguridad informática con la seguridad en la información, al hablar de seguridad de la información se refiere a todo lo que se puede embeber o contener información y no necesariamente en un medio informático. La seguridad de la información se define como un conjunto de medidas preventivas, técnicas y organizativas de una institución que permiten

asegurar y proteger la confidencialidad, integridad y disponibilidad de la información como se indica en la Figura 3.

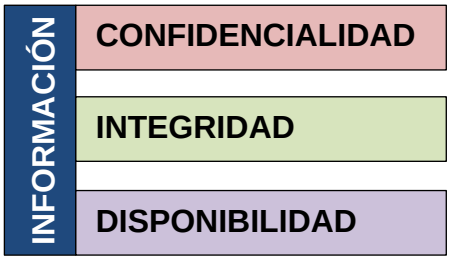

Figura 3: Seguridad de la Información

- **Confidencialidad:** Su objetivo principal es garantizar la propagación de la información a personas no autorizadas. En seguridades informáticas la confidencialidad es proteger los datos de sistemas que deseen acceder a los mismos.
- **Integridad:** Es garantizar que los datos sean los correctos. En seguridades informáticas la integridad es certificar que ningún dato pueda ser modificado al acceder a la información sin la autorización necesaria.
- **Disponibilidad:** Es garantizar la disponibilidad de la información cuando las personas o sistemas autorizadas lo deseen.

2.3.2. Requerimientos básicos para la seguridad de la información

La información que una organización genera es el capital más importante, sin importar el tamaño, giro e infraestructura tecnológica. Conforme una organización va madurando y creciendo, esta premisa es más adoptada con más fuerza y por lo tanto adquiere la importancia que merece.

Todas las organizaciones y usuarios que generan aplicaciones web deben generarlas con un mínimo de seguridad hacia la información que manipulan desde que se comienza a concebir el proyecto pues luego es muy difícil y complicado modificar el sistema una vez ya se encuentra finalizado. Esto conlleva a que se deban tomar ciertas medidas para proteger la información, lo cual deriva en medidas para asegurar las aplicaciones web desarrolladas; de ahí que muchas veces solamente se haga referencia a la seguridad de las aplicaciones web, debido a que la de la información va implícita.

Una serie de requisitos básicos a tener en cuenta para garantizar la seguridad de la información a la vez que se construyen aplicaciones web para brindar servicios son los que se mencionan a continuación:

- **Validación de la entrada y salida de información:** Siempre debe verificarse que cualquier dato entrante o saliente es apropiado y en el formato que se espera. Este es el principal mecanismo del que dispone un atacante para enviar o recibir código malicioso hacia el sistema.
- **Diseños simples:** Los mecanismos de seguridad deben ser los más sencillos posibles, evitando pasos muy complicados para los usuarios y que no sean ejecutados por estos.
- **Utilización y reutilización de componentes de confianza:** Cuando exista un componente que resuelva un problema de forma correcta, debe ser utilizado.

- **Defensa en profundidad:** Se debe contar con mecanismos de seguridad suficientes para que cuando un componente del sistema falle ante un determinado evento, otros sean capaces de detectarlo.
- **Verificación de privilegios:** Los procesos deben contar únicamente con los privilegios necesarios para desarrollar su función, y siempre que sea posible el sistema funcionar con los menos privilegios posibles.
- **Ofrecer la mínima información:** Ante un error o una validación negativa, los mecanismos de seguridad deben facilitar la mínima información posible.

2.3.3. Seguridad informática

Generalmente cuando se habla de seguridad informática se refiere a las características y condiciones que debe cumplir un sistema informático para garantizar la confidencialidad, integralidad y disponibilidad de los datos.

La seguridad informática sirve para proteger la información contra amenazas, para evitar daños y minimizar los riesgos relacionados esta; busca garantizar que los recursos del sistema sean utilizados para el fin que se crearon.

2.3.4. Seguridad en las aplicaciones informáticas

Las aplicaciones web y las de escritorio comparten muchos problemas de seguridad dependiendo de cómo se implementes y de las funciones que necesiten realizar según su negocio; es por eso que muchas veces comparten muchas medidas preventivas, sobre todo cuando las aplicaciones de escritorio necesitan acceder a datos a través de la red. Los desarrolladores deben

capacitarse sobre las vulnerabilidades más comunes y atenderlas durante todo el ciclo de desarrollo.

La seguridad es un aspecto importante para proteger la integridad y privacidad de los datos y recursos de las aplicaciones Web. Siempre se debería designar una estrategia de seguridad para su aplicación Web, que use soluciones de seguridad de eficacia probada, e implementar métodos de autenticación, autorización y validación de datos, para proteger la aplicación de una serie de amenazas.

Se puede tener en cuenta una serie de principios generales de seguridad para las aplicaciones Web, las cuales serían (Zalewski, 2012):

- Mecanismos de autorización: Implementar un mecanismo de autorización potente para restringir el acceso a los recursos y proteger la lógica corporativa.
- Validación de los datos: Utilizar sistemas de validación de datos y validación de entradas en todos los límites de confianza, y así evitar que se exploten errores debidos al procesamiento de datos no válidos. Además de realizar validaciones en el cliente se deben realizar validaciones en el servidor.
- Cifrado de los datos: Cifrar todos los datos importantes que se envíen a través de la red.
- Utilizar servicios web: Implemente la lógica corporativa sensible mediante servicios Web.

- Reducir el tiempo de disponibilidad de los datos: Reducir al mínimo el tiempo que están disponibles los datos importantes en el cliente. Se debe utilizar la carga dinámica de recursos y sobrescribir o borrar los componentes que contengan datos importantes de la caché del navegador.
- Utilización de tokens: Un token de larga duración se puede incluir en la página del lado de cliente para incluirlo en las solicitudes de servicios. Pero no es recomendado utilizar tokens de larga duración más allá del patrón de seguridad SIG básico, dada la posibilidad de repetir ataques en los cuales un atacante intercepta datos y los retransmite (como podría hacer con un token que permitiese el acceso).

2.4. Riesgos y vulnerabilidades

Cuando en informática se menciona que existe un riesgo, esto se refiere a la posibilidad de que una amenaza se produzca, dando lugar a un ataque al equipo o servidor; no es otra cosa que la probabilidad de que ocurra el ataque por parte de la amenaza. El análisis del riesgo existente permite tomar decisiones para proteger mejor al sistema y le permite decidir hasta donde su institución podría aceptarlo.

Las vulnerabilidades son puntos débiles del software que permiten que un atacante comprometa la integridad, disponibilidad o confidencialidad del mismo. Algunas de las vulnerabilidades más severas permiten que los atacantes ejecuten código arbitrario, denominadas vulnerabilidades de seguridad, en un sistema comprometido. Que exista una vulnerabilidad no significa que se vaya a producir un daño de forma automática.

2.4.1. Riesgos de seguridad en aplicaciones Web

Este es uno de los temas más complicados para las empresas que manejan información vital de los visitantes de sus aplicaciones web o sitios web, nombres, direcciones, números de tarjetas de crédito o cualquier otro tipo de información que podría ser utilizada de forma incorrecta y perjudicar a otros.

Pueden generarse muchísimos problemas al tener sistemas informáticos vulnerables en la red, la mayoría de estos se dan por el desconocimiento de las empresas acerca de cuáles son los riesgos y los principales problemas de seguridad que enfrentan sus aplicaciones hoy en día; además de una deficiente validación de la información que se manipula (Parravicini, 2011).

2.4.2. Vulnerabilidades en aplicaciones Web

Las aplicaciones web pueden presentar diversas vulnerabilidades, las cuales ocurren de acuerdo a los servicios que prestan (Aumaille, 2002). De acuerdo con la OWASP (Acrónimo de Open Web Application Security Project, en inglés ‘Proyecto de seguridad de aplicaciones web abiertas’), las vulnerabilidades en aplicaciones web que más han prevalecido son (Palmer, 2011):

1. Inyección.
2. Pérdida de Autenticación y Gestión de Sesiones.
3. Cross-Site Scripting (XSS).
4. Referencias Inseguras a Objetos Directos.
5. Configuración Errónea de Seguridad.
6. Exposición de Datos Sensitivos.
7. Falta de Función de Nivel de Control de Acceso.

8. Falsificación de Requerimientos Cross-Site (CSRF).
9. Utilizar Componentes Vulnerables Conocidos.
10. Redirecciones y reenvíos No Validados.

Esta lista fue iniciada en 2003, cumple 10 años de creada y su propósito es el de crear conciencia sobre la importancia de los riesgos de seguridad que enfrentan las organizaciones que utilizan aplicaciones web.

La fundación OWASP es un proyecto exitoso y está conformado por una serie de guías y proyectos relacionados con la implementación de la seguridad en desarrollos principalmente web. La comunidad OWASP está conformada por empresas, organizaciones educativas y particulares de todo mundo. En su conjunto constituyen una comunidad de seguridad informática que trabaja para crear artículos, metodologías, documentación, herramientas y tecnologías que se liberan y pueden ser usadas gratuitamente por cualquiera.

La gran ventaja de esta comunidad es que no tiene fines de lucro y eso la hace estar libre de presiones corporativas, lo cual facilita que se proporcione información imparcial, práctica y provechosa sobre la seguridad de las aplicaciones informática.

OWASP no está afiliado a ninguna compañía tecnológica, si bien apoya el uso informado de tecnologías de seguridad, recomienda enfocar la seguridad de aplicaciones informáticas considerando todas sus dimensiones: personas, procesos y tecnologías.

Los proyectos OWASP se dividen en dos categorías principales: la primera en proyectos de documentación y la segunda en proyectos de desarrollo los cuales se detallan a continuación.

1. **Proyectos de documentación**

    a. Guía OWASP: Documento que proporciona una guía detallada sobre la seguridad de las aplicaciones web.
    b. OWASP Top 10: Documento de alto nivel que se centra sobre las vulnerabilidades más críticas de las aplicaciones web.
    c. Métricas: Proyecto para definir métricas aplicables de seguridad de aplicaciones web.
    d. Legal: Un proyecto para ayudar a los vendedores y compradores de software a negociar adecuadamente los aspectos de seguridad en sus contratos.
    e. Guía de pruebas: Guía centrada en la prueba efectiva de la seguridad de aplicaciones web.
    f. ISO/IEC 27002 (anteriormente denominada ISO 17799): Documentos de apoyo para organizaciones que realicen revisiones ISO/IEC 27002.
    g. AppSec FAQ: Preguntas y respuestas frecuentes sobre seguridad de aplicaciones web.

La ISO/IEC 2700, es una guía de buenas prácticas que describe los objetivos de control y controles recomendables en cuanto a seguridad de la información. Va orientado a la seguridad de la información, de modo que las probabilidades de ser afectados por daño, robo o pérdida de información se vean minimizados al máximo. Contiene un número de categorías de seguridad principales, entre las cuales se tienen 11 cláusulas. (ISO, 2013)

2. **Proyectos de desarrollo**

    a. WebScarab: Aplicación de chequeo de vulnerabilidades de aplicaciones web incluyendo herramientas proxy.

    b. Filtros de validación (Stinger para J2EE, filters para PHP): Son filtros genéricos de seguridad perimetral que los desarrolladores pueden usar en sus propias aplicaciones.

    c. WebGoat: Herramienta interactiva de formación y benchmarking para que los usuarios aprendan sobre seguridad de aplicaciones web de forma segura y legal.

    d. DotNet: Conjunto de herramientas para explorar y mejorar la seguridad en los entornos NET.

Construir software inseguro tiene sus consecuencias, las aplicaciones web inseguras, expuestas a millones de usuarios a través de Internet, representan una inquietud creciente. Incluso, la confianza de los clientes que utilizan la Web para realizar sus compras o cubrir sus necesidades de información está decreciendo, a medida que más y más aplicaciones web se ven expuestas a ataques.

2.5. OWASP Top 10 – 2010

Los atacantes pueden potencialmente utilizar muchas rutas diferentes a través de su aplicación para causar daño en su negocio u organización. Cada una de estas rutas representa un riesgo que puede, o no, ser lo suficientemente serio como para ser atendido.

En muchas ocasiones estas rutas son fáciles de encontrar y explotar y otras veces son extremadamente difíciles. De la misma forma, el daño causado

puede ir de ninguno hasta incluso sacarlo del negocio. Para determinar el riesgo para su organización, puede evaluar la probabilidad asociada con cada agente de amenaza, vector de ataque y debilidad de seguridad y combinarla con una estimación del impacto técnico y de negocios en su organización. Juntos, estos factores determinan el riesgo total.

La actualización del OWASP Top 10 del año 2010, se enfoca en la identificación de los riesgos más serios para un amplio espectro de organizaciones. Para cada uno de estos riesgos, se provee información genérica acerca de su significado y el impacto técnico.

Aunque las versiones previas del OWASP Top 10 se enfocaron en la identificación de las “vulnerabilidades” más comunes, también fueron diseñadas alrededor de los riesgos. Los nombres de los riesgos en la Top 10 surgen del tipo de ataque, el tipo de debilidad o el tipo de impacto que pueden causar. A continuación se expone cada uno de los riegos y se hará mención a su posible impacto.

2.5.1. Inyección

Las fallas de inyección, tales como SQL, OS, y LDAP, ocurren cuando la aplicación permite que datos no confiables sean enviados a un intérprete como parte de un comando o consulta. Los datos hostiles del atacante pueden engañar al intérprete y ejecutar comandos no intencionados o acceder datos no autorizados.

Por lo general su impacto es severo, como lo muestra la Figura 4. Todos los contenidos de una base de datos pueden potencialmente ser leídos o modificados. También puede permitir el completo acceso al esquema de la

base de datos, o cuentas de usuario, o incluso a nivel del sistema operativo. Un escenario muy claro de este ataque es cuando la aplicación utiliza datos no confiables al momento de construir una consulta SQL.

Figura 4: Inyección. Fuente: (OWASP, 2014).

2.5.2. Secuencia de Comandos en Sitios Cruzados (XSS)

Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso.

Típicamente permite robar la sesión del usuario, robar datos sensibles, redireccionar un usuario hacia un sitio de malware o phishing. El impacto es mucho más grave como lo indica la Figura 5, si el atacante instala un proxy XSS que le permita observar y dirigir todas las actividades de un usuario en el sitio vulnerable y forzarlo hacia otros sitios.

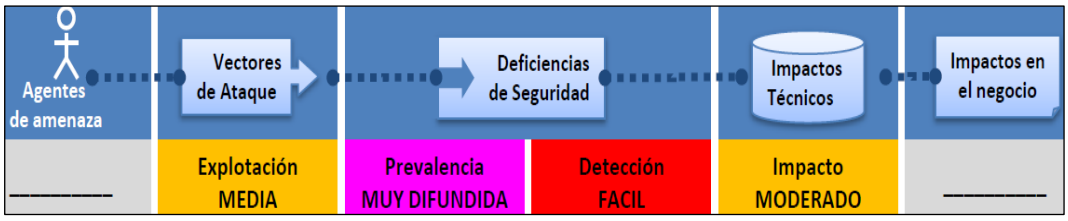

Figura 5: Secuencias de Comandos en Sitios Cruzados. Fuente: (OWASP, 2014).

2.5.3. Pérdida de Autenticación y Gestión de Sesiones

Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente, permitiendo a los atacantes comprometer contraseñas, llaves, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios.

El impacto es severo como lo muestra la Figura 6 y radica en que se comprometen cuentas de usuarios o las sesiones de usuario son secuestradas.

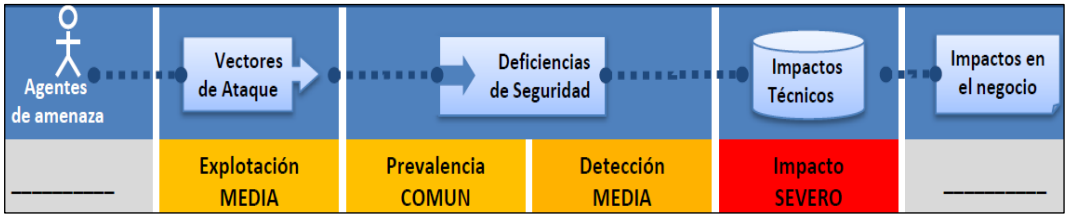

Figura 6: Pérdidas de Autenticación y Gestión de Sesiones. Fuente: (OWASP, 2014).

2.5.4. Referencia Directa Insegura a Objetos

Una referencia directa a objetos ocurre cuando un desarrollador expone una referencia a un objeto de implementación interno, tal como un fichero, directorio, o base de datos, sin un chequeo de control de acceso u otra protección, lo cual permite que los atacantes pueden manipular estas referencias para acceder datos no autorizados. El impacto típico radica en que usuarios son capaces de acceder ficheros o datos sin autorización.

Figura 7: Referencia Directa Insegura a Objetos. Fuente: (OWASP, 2014).

2.5.5. Falsificación de Peticiones en Sitios Cruzados (CSRF)

Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificada, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida de forma automática, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima a generar pedidos que la aplicación vulnerable procesa como si fueran peticiones legítimas provenientes de la víctima.

Este fallo permite iniciar transacciones (transferencia de fondos, desconectar el usuario, cierre de cuenta, etc), acceder a datos sensitivos y cambiar detalles de la cuenta, el impacto alcanzado es moderado como lo muestra la Figura 8.

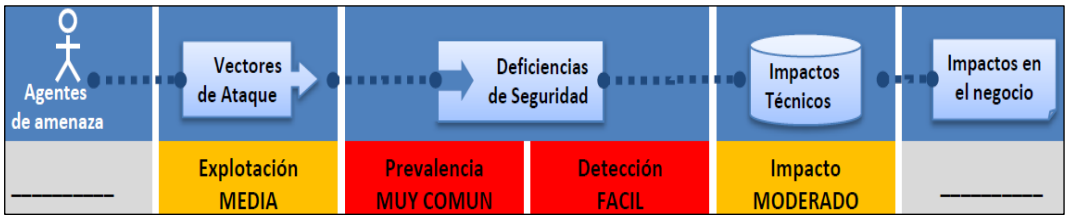

Figura 8: Falsificación de Peticiones en Sitios Cruzados (CSRF). Fuente: (OWASP, 2014).

2.5.6. Defectuosa Configuración de Seguridad

Una buena seguridad requiere tener definida e implementada una configuración segura para la aplicación, marcos de trabajo, servidor de aplicación, servidor web, base de datos, y plataforma. Estas configuraciones deben ser definidas, implementadas, y mantenidas ya que por lo general no son seguras por defecto. Esto incluye mantener todo el software actualizado, incluidas las librerías de código utilizadas por la aplicación.

El impacto es moderado como lo muestra la Figura 9, se basa en la

instalación de código malicioso debido a desactualizaciones, parches faltantes en el Sistema Operativo o servidor, falla de XSS debido parches faltantes en el framework de la aplicación, acceso no autorizado a cuentas por defecto, funcionalidad de la aplicación, etc, debido a una defectuosa configuración del servidor.

Figura 9: Defectuosa Configuración de Seguridad.
Fuente: (OWASP, 2014).

2.5.7. Almacenamiento Criptográfico Inseguro

La criptografía es una técnica que se utiliza para proteger documentos y datos. Funciona a través de la utilización de cifras o códigos para escribir algo de forma secreta en documentos y datos confidenciales y permitirles circular de forma segura, evitando la lectura por personas no autorizadas.

Para acceder a la información encriptada se debe poseer una llave, la cual está implicada en el proceso de cifrado/descifrado y puede ser o no igual a la del emisor dependiendo del sistema de cifrado empleado. Los sistemas de cifrados puedes ser simétrico, asimétrico e híbrido.

Muchas aplicaciones web no protegen adecuadamente los datos sensibles, tales como tarjetas de crédito, NSSs, y credenciales de autenticación con mecanismos de cifrado o hashing. Esto facilita que los atacantes puedan modificar o robar los datos que se encuentran desprotegidos para conducir robos de identidad, fraudes de tarjeta de crédito u otros crímenes.

El mayor impacto de este fallo de seguridad radica en que permite que los atacantes accedan o modifican información privada o confidencial, por ejemplo tarjetas de crédito, registros médicos, datos financieros; permite a los atacantes extraer secretos a ser usados en otros ataques, mala Imagen para la Compañía, clientes insatisfechos, y pérdida de confianza, gastos para corregir el incidente, tales como análisis forense, enviar cartas de disculpas, reemisión de tarjetas de crédito, etc., el Negocio es demandado o multado y el impacto técnico es severo como lo muestra la Figura 10.

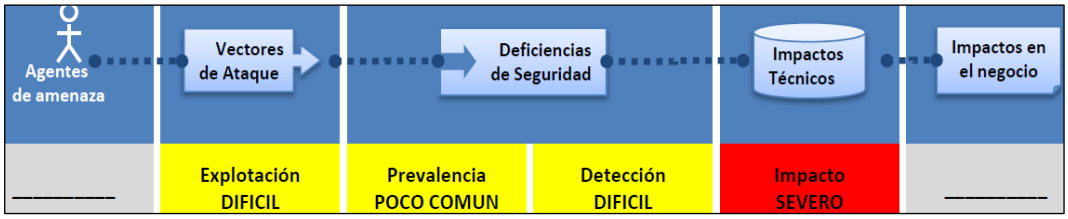

Figura 10: Almacenamiento Criptográfico Inseguro.
Fuente: (OWASP, 2014).

2.5.8. Falla de Restricción de Acceso a URL

Una práctica muy empleada y de fácil implementación resulta la verificación de los privilegios de acceso a URLs antes de generar enlaces o botones protegidos. Para ello, las aplicaciones necesitan realizar controles similares cada vez que estas páginas son accedidas, o los atacantes podrían falsificar URLs para acceder a estas páginas igualmente.

Su impacto técnico es moderado como lo muestra la Figura 11 y el explotación de esta falla permitiría que los atacantes invocaran funciones y servicios a los cuales no se encuentran autorizados, el acceso a otras cuentas de usuario y datos, a realizar acciones privilegiadas o de administración, etc.

Un ejemplo muy común es cuando la aplicación utiliza datos no verificados

mediante un intérprete SQL e ingresa a la información deseada.

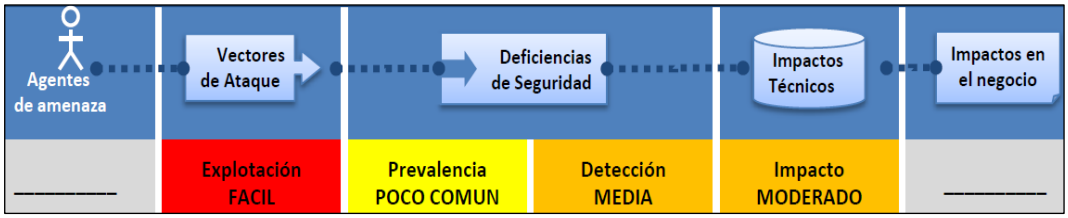

Figura 11: Falla de Restricción de Acceso a URL. Fuente: (OWASP, 2014)

2.5.9. Protección Insuficiente en la Capa de Transporte

Las aplicaciones frecuentemente fallan al autenticar, cifrar, y proteger la confidencialidad e integridad de tráfico de red sensible. Cuando esto ocurre, es debido a la utilización de algoritmos débiles, certificados expirados, inválidos, o sencillamente no utilizados correctamente.

La Figura 12 muestra como este error puede ser aprovechado por un atacante y acceder o modificar información privada o confidencial, por ejemplo, tarjetas de crédito, registros médicos, datos financieros; extraer secretos a ser usados en otros ataques, transmitir una mala imagen para la Compañía, generar desconfianza en la compañía, generar gastos para corregir el incidente, etc.

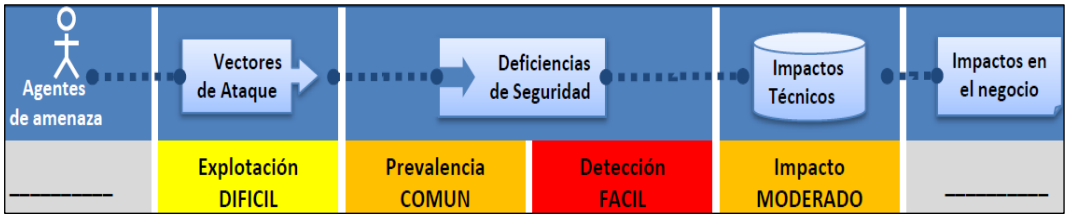

Figura 12: Protección Insuficiente en la Capa de Transporte. Fuente: (OWASP, 2014)

2.5.10. Redirecciones y reenvíos no validados

Las aplicaciones web frecuentemente redirigen y reenvían a los usuarios

hacia otras páginas o sitios web, y utilizan datos no confiables para determinar la página de destino. Sin una validación apropiada, los atacantes pueden redirigir a las víctimas hacia sitios de phishing o malware, o utilizar reenvíos para acceder páginas no autorizadas.

Su impacto técnico es moderado como lo muestra la Figura 13 y un fallo de esta índole provocaría que el pedido del atacante sea ejecutado, pasando por alto los controles de seguridad.

Figura 13: Redirecciones y reenvíos no validados. Fuente: (OWASP, 2014)

CAPÍTULO 3

ANALISIS DE LAS APLICACIONES WEB DE LA SBS

3.1. Situación Actual de las aplicaciones Web

Con el transcurso del tiempo los sistemas bancarios han evolucionado e incorporado a su funcionamiento sistemas informáticos debido al desarrollo de las Tecnologías de la Información y la Comunicación (TIC). Por consiguiente garantizar la seguridad en las aplicaciones web de los sistemas bancarios es sumamente necesario, pues de ello depende la integridad, confidencialidad y disponibilidad de la información almacenada en distintos equipos de la red, para evitar así problemas tecnológicos como: la vulnerabilidad y los riegos que podrían ocasionar grandes pérdidas financieras. La SBS definió la plataforma tecnológica que deben utilizar los sistemas actuales, por lo cual todas las herramientas que se desarrollen deben guiarse por la arquitectura definida por la empresa. Se definió que el acceso a los sistemas de la SBS debe ser a través de los navegadores de internet: Internet Explorer versión 7.0 o superior y Firefox 4.0 o superior. Las herramientas deben permitir el acceso a menús, pantallas, reportes u otras partes de la misma únicamente a usuarios autenticados y autorizados; y cada una debe mantener logs de auditoría y acceso. Además no se permitirá el acceso directamente a través de URL's.

Como sistema de gestión de base de datos objeto-relacional se utiliza Oracle, el cual se considera uno de los sistemas de bases de datos más completos, debido a las características que brinda: soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma. Para facilitar la gestión de

los sistemas se utiliza Open LDAP que es una implementación libre y de código abierto del protocolo ligero de acceso a directorios.

Se utiliza Java Enterprise Edition (JEE) para desarrollar y ejecutar software de aplicaciones con el lenguaje de programación Java. Se emplea JBoss Developer Studio como plataforma empresarial Java también de código abierto. El marco de aplicación web que se utiliza para implementar las interfaces de usuario es Java Server Faces (JSF) lo cual hace más fácil para los desarrolladores conectar el nivel de presentación con el código de la aplicación. Se hace uso de Enterprise Java Beans (EJB) una de las API que forman parte del estándar de construcción de aplicaciones empresariales JEE y para combinar los frameworks EJB y JSF se usa JBoss Seam. Como entorno de desarrollo integrado (IDE) desarrollado por Oracle Corporation se emplea JDeveloper, el cual soporta diferentes lenguajes como Java, HTML, XML, SQL, PL/SQL, JavaScript, PHP, Oracle ADF y UML. Como entorno de desarrollo integrado (IDE) se utiliza Eclipse, compuesto por un conjunto de herramientas de programación de código abierto y multiplataforma. (Paredes, 2014), (Erodata, 2012a), (SBS, 2013)

3.2. Sistemas más importantes de la SBS con tecnología JEE

La Superintendencia de Bancos y Seguros continúa optimizando sus procesos tecnológicos los cuales brindan varias facilidades a las entidades controladas. A continuación se presentan algunos de los sistemas más importantes de la SBS que utilizan JEE, dentro de los cuales se encuentran:

- Sistema de Población de Identificaciones (SPI)
- Sistema de Auditoría de Prevención de Lavado de Activos (SAPLA)

- Sistema para otorgar Credenciales a Intermediarios de Seguros (SOCI)
- Herramienta de Apoyo al Manual Único de Supervisión (HAMUS)

De cada uno de estos sistemas se especifica el objetivo para el cual fue creado y algunas de las características que presentan.

3.2.1. Sistema de Población de Identificaciones (SPI)

El Sistema de Población de Identificaciones (SPI) es una aplicación web que permite el acceso a la información de personas naturales y/o jurídicas ecuatorianas así como también para personas naturales y/o jurídicas extranjeras, que tengan permisos para acceder al sistema.

Permite consultar, poblar, modificar y eliminar registros de personas naturales ecuatorianas así como también personas naturales y/o jurídicas extranjeras. Permite además consultar, poblar registros de personas jurídicas ecuatorianas. También es posible obtener reportes de población, modificación, eliminación y consultas realizadas a la DIGERCIC. El sistema está compuesto por 5 menús, los cuales son:

- Consulta de identificaciones.
- Población de identificaciones.
- Modificación de identificaciones.
- Eliminación de identificaciones y Reportes. (Ecuador, 2013), (Armijos Tandazo, 2012)

3.2.2. Sistema de Auditoría de Prevención de Lavado de Activos (SAPLA)

El sistema de Auditoría de Prevención de Lavado de Activos (SAPLA) fue concebido y solicitado por la Subdirección de Prevención de Lavado de Activos

(SPLA) de la Superintendencia de Bancos y Seguros del Ecuador. La empresa que se encargó de diseñarlo y desarrollarlo fue la empresa ecuatoriana SASCURE CÍA. LTDA.

El sistema SAPLA da soporte a diferentes tipos de auditorías, dentro de las cuales se encuentran: las auditorías integrales, auditorías focalizadas o las auditorías de seguimiento. Además incluye la aplicación de procedimientos que facilitan la evaluación para instituciones financieras, empresas de seguros y compañías de reaseguros.

Una de las mayores ventajas del sistema SAPLA es la automatización del seguimiento de los planes de cumplimiento establecidos en cada informe de auditoría. Dicho procedimiento se realiza a través de correos electrónicos emitidos por el sistema en forma automática en una determinada fecha de cada mes hacia los responsables del cumplimiento, a manera de recordatorio, con el fin de solicitarles el descargo de las recomendaciones efectuadas por los auditores de la SBS. Esta automatización evita a SAPLA realizar esta tarea de forma manual, para las más de 150 instituciones que supervisa y de manera personalizada según cada plan de acción acordado.

Adicionalmente el sistema SAPLA permite obtener reportes por institución, por subsistema y reportes gerenciales que permiten realizar análisis comparativos del estado de las auditorías en cualquier fecha, clasificadas por diferentes variables de comportamiento, así como de la evolución del cumplimiento de las recomendaciones efectuadas. El ciclo de vida de las auditorías a las que da soporte el sistema SAPLA son las siguientes: Planificación de las auditorías, Ejecución de las auditorías, Emisión de los

reportes, Seguimiento y finalmente Generación de un reporte final. El sistema está compuesto por 6 módulos: Administración, Planificación, Ejecución, Seguimiento, Reportes y Seguridad. Cuenta con un máximo de 2 usuarios administradores del sistema y un mínimo de 1 usuario supervisor. Adicionalmente se pueden considerar otros perfiles de consulta. El objetivo del sistema SAPLA es:

- Proporcionar al usuario una herramienta informática que permita el manejo de las auditorías de manera sencilla, ágil y parametrizada.
- Mantener una organización adecuada y consolidada de la información de las auditorías.
- Proporcionar información gerencial, administrativa y de seguridad de las auditorias efectuadas.
- Facilitar al usuario el seguimiento del cumplimiento de los planes de acción por parte de las entidades controladas por la Subdirección de Prevención de Lavado de Activos. (Sascure CIA, 2012)

3.2.3. Sistema Otorgar Credenciales a Intermediarios de Seguros (SOCI)

El sistema SOCI es un sistema basado en web que permite la administración del proceso de solicitud y rendición de exámenes previos a la obtención de un certificado para ser intermediario de seguros. Una persona natural o jurídica que quiera obtener un certificado en alguna rama de seguro debe realizar una solicitud en línea a través del sistema, en la cual se solicitan los requisitos necesarios para que sea apto y entonces pueda rendir el examen correspondiente.

La solicitudes son procesadas por miembros de la SBS en la cual se define si cumple o no los requisitos. Una vez aprobadas o rechazadas se notifica al aspirante la fecha en la cual rendirá el examen o la causa de rechazo de ser el caso. El aspirante rendirá el examen en el día y hora especificado, para lo cual se genera una prueba con preguntas aleatorias sacadas de un banco de preguntas alimentadas por la SBS.

Al final del tiempo establecido o finalizado el examen se obtienen los resultados del mismo y el administrador del sistema emite el resumen correspondiente al proceso. Para administrar estos macro procesos y subprocesos el sistema cuenta con tres módulos principales: Módulo de Gestión, Módulo del Aspirante y Módulo del Examen. (ERODATA, 2012)

3.2.4. Herramienta de Apoyo al Manual Único de Supervisión (HAMUS)

Esta herramienta permite dinamizar y optimizar los procesos de supervisión efectuados por la Intendencia Nacional del Sector Financiero Privado. El enfoque del MUS contempla una supervisión integral, basada en riesgos y con una permanente actualización y desarrollo, que incluye el mantenimiento y almacenamiento de las bases de datos generadas en las supervisiones concluidas y en curso, las mismas que aportan a:

- Una regulación efectiva y prudente.
- Un sistema de supervisión continua que contempla el análisis a distancia y de las inspecciones en campo.
- Diseño y coordinación de la implementación del sistema de información de supervisión para la aplicación del manual.

Los objetivos específicos de la herramienta son los siguientes:

- Registrar las supervisiones concluidas y las correspondientes calificaciones GREC por componente y subcomponente desde el año 2010 y en adelante.
- Crear y mantener actualizado un repositorio de documentos que almacene la información de:
    - Supervisiones históricas desde el año 2010.
    - Planificaciones (entidades, alcance, fecha de inicio y fecha de finalización).
    - Tareas con estado (en curso o concluidas), responsable y fecha de cumplimiento.
    - Fichas de calificación por cliente efectuadas en las inspecciones in-situ.
    - Calificaciones GREC historias de las entidades supervisadas.
    - Expedientes por funcionario, que permita identificar el grado de conocimientos, destrezas y funciones dentro de la INSFPR.
- Proveer de una herramienta informática flexible, que permita soportar el proceso de regulación y supervisión establecido.
- Proveer un servicio en la calidad, disponibilidad e integridad de la información relacionada a la supervisión. (Bupartech, 2013)

Después de haber explicado en qué consisten cada uno de los sistemas se decide analizar 3 de las 4 aplicaciones, debido a que son algunos de los sistemas más complejos de la SBS y que manejan gran cantidad de

información. Además las acciones que realizan son claves o esenciales para el correcto funcionamiento de la SBS. Igualmente son sistemas que constantemente actualizan sus datos por lo que necesitan tener buena seguridad para mantener su integridad. Los sistemas a los que se le va a aplicar el Top 10 de OWASP son: Sistema de Población de Identificaciones (SPI), Sistema de Auditoría de Prevención de Lavado de Activos (SAPLA) y Sistema para otorgar Credenciales a Intermediarios de Seguros (SOCI).

3.3. Arquitectura de las aplicaciones Web de la SBS

Una arquitectura es el conjunto de decisiones sobre la organización de la aplicación web que define los principios que guían el desarrollo, los componentes principales del sistema, responsabilidades y la forma en que se interrelacionan. JEE es un estándar para el desarrollo de aplicaciones empresariales (portables, robustas, escalables y seguras) que usa tecnología Java. La mayoría de los productos que realizan la SBS utilizan la arquitectura Java Enterprise Edition (JEE) para el desarrollo de sus aplicaciones web. Como se muestra en la Figura 14, JEE incluye varias especificaciones de API o interfaz de programación de aplicaciones, como son JDBC, RMI, e-mail, JMS, Servicios Web, XML. Además también configura algunas especificaciones únicas de JEE para componentes. Estas incluyen Enterprise JavaBeans, servlets, portlets y varias tecnologías de servicios web tales como Axis2 o JAX-WS. Uno de los beneficios que brinda es que el servidor de aplicaciones puede manejar transacciones, seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados. Dicha plataforma brinda varios servicios como son:

- Alta disponibilidad, lo cual posibilita que el servicio sea usado sin problemas la mayor parte del tiempo.
- Seguridad para asegurar la privacidad de los usuarios, además de la integridad y confidencialidad de las transacciones y la información procesada.
- Escalabilidad, la cual garantiza que los servicios sigan siendo operativos aunque el número de usuarios, de transacciones o el volumen de información sufran aumentos importantes.

Las ventajas que proporciona son las siguientes:

- Soporte para múltiples plataformas y sistemas operativos.
- Avalado por múltiples empresas (SUN, IBM, ORACLE).
- Competitividad.
- Soluciones libres.

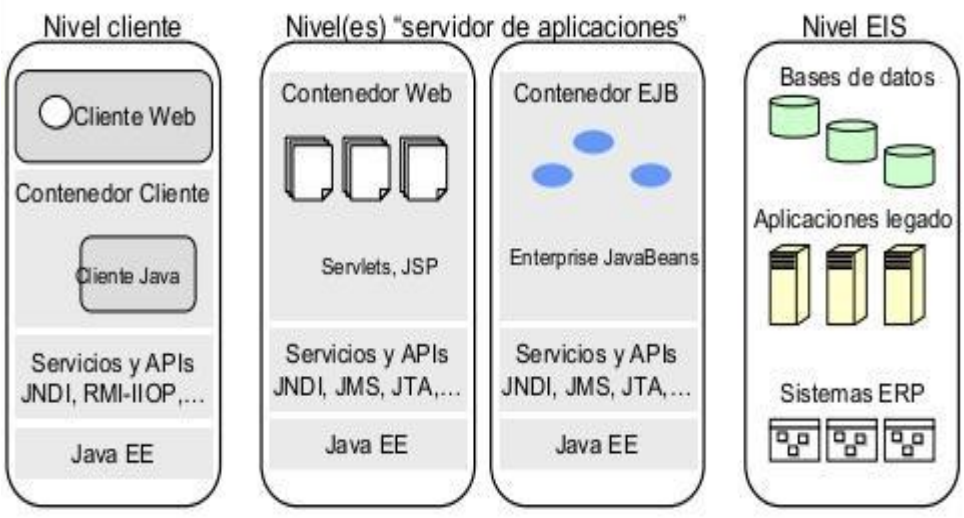

Figura 14: Arquitectura Java Enterprise Edition.
Fuente: (Pickin & et al, 2010).

En los niveles de los servidores de aplicaciones se utilizan los Contenedores (containers), a continuación se mencionan algunas de sus principales características:

- Ofrecen un entorno de ejecución para los componentes de la aplicación.
- Proporcionan una vista uniforme de los servicios requeridos por ellos especificados en los descriptores (especificaciones de dependencias).
- Proporcionan herramientas de despliegue para la instalación y configuración de componentes.
- Las tareas principales de los contenedores del lado del servidor son la gestión de recursos y del ciclo de vida. (Pickin & et al, 2010)

3.3.1. Enterprise JavaBeans

Enterprise Java Beans (EJB) es una completa especificación de arquitectura para componentes de servicio. Facilita el desarrollo de aplicaciones, concentrándose en la lógica de negocio, desarrollo, aplicación y aspectos de tiempo de ejecución. Logra la independencia del proveedor de componentes mediante la especificación de interfaces y de la plataforma gracias al principio Write Once Run Anywhere (WORA) y a su realización en Java. Asegura la compatibilidad con el Java-Apis existente, con los sistemas de servidor de terceros y con protocolos de CORBA y de servicios Web. (Pickin & et al, 2010)

Algunos de los componentes disponibles en la arquitectura Java Enterprise Edition para el encapsulamiento de la lógica del negocio, son los siguientes:

- Enterprise Java Beans.
- Session Beans.

- Mensajería / asincronismo.
- Message Driven Beans.

3.3.2. Servicios JEE

- Servicio de nombres: acceso a componentes y recursos mediante nombres lógicos.
- Servicio de transacciones ejecución de una serie de pasos de forma atómica y aislada: Demarcación de transacciones programáticas: posibilidad de un control de las transacciones mediante una interfaz de programación Java Transaction Service (JTS).
- Servicio de seguridad: directivas de seguridad para recursos protegidos. Control de acceso en dos pasos: autentificación y autorización. Realización declarativa o programada. (Pickin & et al, 2010)

3.4. Identificación de riesgos en base al Top 10 de OWASP

El Proyecto Abierto de Seguridad de Aplicaciones Web (OWASP), es una comunidad abierta que le permite a las organizaciones realizar el desarrollo y mantenimiento de las aplicaciones. Todas las herramientas, documentos, foros y delegaciones de OWASP son libres y abiertos a las personas que deseen mejorar la seguridad de las aplicaciones. El proyecto OWASP es un nuevo tipo de organización que puede facilitar información imparcial, práctica y ajustada en costes sobre la seguridad de las aplicaciones. OWASP no está asociado a ninguna compañía tecnológica, aunque puede dar soporte a la información de uso de tecnología de seguridad comercial. Similar a varios proyectos de código

abierto, OWASP produce materiales con un desarrollo colaborativo. (OWASP.org, 2014).

En la actualidad los sistemas informáticos inseguros causan daños significativos a diferentes áreas de la sociedad, como la salud, defensa, finanzas y energía. A medida que avanza la era digital se hace cada vez más complejo lograr aplicaciones seguras. OWASP creó un Top 10 con el objetivo de crear conciencia sobre la seguridad de las aplicaciones mediante la identificación de algunos de los riesgos más críticos que enfrentan las organizaciones. El objetivo principal del Top 10 es educar a desarrolladores, diseñadores, arquitectos, gerentes y a organizaciones sobre las consecuencias de las vulnerabilidades de seguridad en aplicaciones web. Además el Top 10 provee técnicas básicas sobre cómo proteger las áreas de alto riesgo y provee también orientación sobre los pasos a seguir. (OWASP.org, 2014)

Durante el desarrollo de un software es importante descubrir las vulnerabilidades que presenta y estimar el riesgo asociado para el negocio. En las primeras etapas del ciclo de vida del desarrollo del software se pueden identificar cuestiones que afecten a la seguridad en la arquitectura o diseño, mediante el uso de técnicas de modelado de amenazas. Más adelante, se pueden encontrar problemas de seguridad realizando revisiones de código o pruebas de intrusión. O simplemente no se descubre ningún problema hasta que la aplicación esté en producción y ya ha sido comprometida su seguridad

Existen varios enfoques para realizar un análisis de riesgos, en esta investigación se utiliza el enfoque de OWASP basado en varias metodologías, entre las que se encuentra la metodología de valoración de riesgos. Esta

metodología plantea que el primer paso es identificar un riesgo de seguridad que necesite ser valorado. Se necesita recopilar información sobre los agentes que causan la amenaza, el ataque que utilizan, la vulnerabilidad involucrada y el impacto de una explotación con éxito del negocio. A continuación en la Tabla 1 se realiza un análisis de cada uno de los sistemas en base al TOP 10 de OWASP. (OWASP, 2014).

Tabla 1: Análisis de los sistemas en base al TOP 10 de OWASP

	SAPLA	SPI	SOCI
R1	Se utilizan consultas estáticas y variables parametrizadas. Ver Anexos 1, 2, 3.		
R2	Validan los datos de entrada y las peticiones HTTP para cada sesión. Ver Anexos 4, 5, 6.		
R3	Si los usuarios no cierran las sesiones, éstas caducan a los 30 minutos de inactividad.		
R4	Se definen los permisos sobre los menús o perfiles que tiene cada usuario. Ver Anexos 7, 8.		
R5	Cada enlace, sesión y formulario, contiene un token de seguridad no predecible para los usuarios. Ver Anexos 9, 10.		
R6	Se trabaja con un servidor web Jboss el cual mantiene subido un firewall que no permite el acceso a la consola de administración mediante su IP. Ver Anexos 11, 12, 14.		
R7	No utilizan ningún algoritmo para encriptar la información		Se encriptan datos con Advanced Encryption Standard y MD5 para encriptar las contraseñas. Ver Anexos 16,17
R8	Se emplean mecanismos de seguridad para el acceso a las páginas, mediante la autenticación y autorización. Ver Anexos 9, 13, 15.		
R9	No se utiliza un protocolo de conexión segura como SSL.		
R10	Las redirecciones y los reenvíos están validados. Ver Anexos 9, 10.		

A partir de la comparación y el análisis realizado se identificaron fundamentalmente la ocurrencia de los siguientes riesgos:

- R7: Almacenamiento Criptográfico Inseguro.
- R9: Protección Insuficiente en la Capa de Transporte.

En el caso del sistema SOCI utiliza el algoritmo hash MD5 el cual es uno de los que más se utiliza hoy en día. Este algoritmo se considera débil y se recomienda que sea reemplazado en el futuro por otro algoritmo de encriptación que sea más seguro. En el caso de los sistemas SAPLA y SPI no se utiliza ningún algoritmo para encriptar los datos, lo que constituye una vulnerabilidad considerable, pues si se accede a los datos, se encontrarían en texto plano, por lo que sería más fácil para los atacantes realizar un ataque exitoso a la aplicación.

El riesgo Protección insuficiente en la capa de transporte está presente en las aplicaciones, pues en ninguna de ellas se utiliza un protocolo criptográfico como SSL (Secure Sockets Layer, capa de conexión segura) para proteger todo el tráfico relacionado con la autenticación. Tampoco se utiliza para encriptar los canales de comunicación de datos, los servicios y los recursos. La utilización de este protocolo facilitaría la protección, confidencialidad y autenticación de la información transmitida.

Aunque el riesgo Inyección no está presente en las aplicaciones analizadas anteriormente, porque se utilizan variables parametrizadas en consultas estáticas. Se recomienda utilizar la función PreparedStatement de Java, debido a que es la primera opción de defensa que propone OWASP para prevenir ataques de inyección.

Aunque el riesgo Pérdida de Autenticación y Gestión de Sesiones no está presente en las aplicaciones analizadas, se recomienda que se reduzca el tiempo de cierre de las sesiones. Actualmente este tiempo se configura en el CAS (Central Authentication Service) y la sesión se cierra 30 minutos después de que no se registre actividad por parte del usuario que está autenticado, este es un tiempo extenso por lo que se aconseja que el mismo se reduzca a 15. De esta forma se evita que los atacantes tengan menos oportunidad de realizar un ataque en ese tiempo.

Con el objetivo de estimar la gravedad de los riesgos del Top 10 de OWASP se utiliza la metodología de evaluación de riesgos. En este caso se aplica la metodología para evaluar los dos riesgos identificados en las aplicaciones analizadas. Para aplicar la metodología se utiliza la guía que aparece en la página oficial de OWASP, específicamente el artículo Risk Rating Methodology. (OWASP.org, 2014).

3.5. Factores de Riesgo para la estimación

Después de identificar un riesgo potencial, se debe estimar la “probabilidad de ocurrencia”. Esta es una medida aproximada de la probabilidad de que la vulnerabilidad sea descubierta y explotada por un atacante. No es necesario ser precisos en esta estimación, es suficiente identificar si la probabilidad de ocurrencia es baja, media o alta.

Existen varios factores que permiten realizar esta estimación. El primer grupo de factores está relacionado con los agentes que causan la amenaza involucrada. El objetivo es estimar la probabilidad de ocurrencia de un ataque con éxito por parte de un grupo de posibles atacantes. (OWASP, 2012) Cada

factor tiene asociado un conjunto de opciones y cada opción tiene asociada un número que está entre 0 y 9 que significa la probabilidad de ocurrencia. Después estos valores su utilizan para estimar la probabilidad de ocurrencia global de los riesgos identificados. A continuación se especifica cuál es la probabilidad de ocurrencia:

- De 0 a < 3 la probabilidad se califica como bajo.
- De 3 a < 6 la probabilidad se califica como medio.
- De 6 a 9 la probabilidad se califica como alto.

3.5.1. Factores relacionados con el agente causante de la amenaza

El primer conjunto de factores se relaciona con el agente que origina la amenaza. Por lo tanto el objetivo es estimar la probabilidad de ocurrencia de que un ataque se lleve a cabo con éxito por un grupo de atacantes. Se pueden considerar agentes causantes de las amenazas cualquier persona que pueda enviar datos no confiables al sistema, incluyendo usuarios externos, internos, administradores y empleados con acceso privilegiado. También se pueden considerar atacantes usuarios del sistema, los cuales podrían intentar robar cuentas de otros usuarios. Considerar también a trabajadores que quieran enmascarar sus acciones, así como usuarios con contraseñas auténticas que puedan ser utilizadas para comprometer el sistema. Además pudiera ser cualquier persona que pueda suplantar a usuarios en el momento de enviar peticiones a al sitio web. Otro agente de amenaza puede ser cualquier sitio web u otros canales HTML, a los cuales accedan los usuarios de un determinado sitio web. De igual forma se puede considerar la probabilidad de que alguien

pueda capturar el tráfico de red de sus usuarios. A continuación se mencionan algunos de los factores relacionados con el agente causante de la amenaza.

- **Nivel de conocimiento**: Se indica los conocimientos técnicos que tiene un grupo de atacantes.
    - Sin conocimientos (1).
    - Algunos conocimientos técnicos (3).
    - Usuario avanzado de ordenador (4).
    - Conocimientos de redes y programación (6).
    - Conocimientos de intrusiones de seguridad (9).
- **Motivación:** Motivación de este grupo de atacantes para encontrar y explotar esta vulnerabilidad.
    - Baja motivación o ninguna recompensa (1).
    - Posible recompensa (4).
    - Recompensa alta (9).
- **Oportunidad:** Oportunidades tiene este grupo de atacantes de encontrar y explotar esta vulnerabilidad.
    - Ningún acceso conocido (1).
    - Acceso limitado (4).
    - Acceso total (9).
- **Tamaño:** Numerosidad del grupo de atacantes.
    - Desarrolladores (8).
    - Administradores de sistemas (8).
    - Usuarios de la intranet (4).

- Socios (5).
- Usuarios autenticados (6).
- Usuarios anónimos de Internet (1).

3.5.2. Factores que afectan a la vulnerabilidad

El siguiente conjunto de factores están relacionados con la vulnerabilidad en cuestión. El objetivo es estimar la probabilidad de que la vulnerabilidad sea descubierta y explotada.

- **Facilidad de descubrimiento**: Factibilidad para descubrir esta vulnerabilidad.
  - Prácticamente imposible (9).
  - Difícil (5)
  - Fácil (2).
  - Existen herramientas automatizadas disponibles (1).
- **Facilidad de explotación:** Facilidad para los atacantes de explotar esta vulnerabilidad.
  - Fácil (9).
  - En teoría es posible explotarla (5).
  - Difícil (3).
- **Conocimiento de la vulnerabilidad:** Conocimiento sobre una vulnerabilidad muy conocida.
  - Desconocida (9)
  - Oculta (6)
  - Obvia (4)
  - Se conoce de forma pública (1)

- **Detección de la intrusión:** Frecuencia se detecta un exploit.
    - Detección activa en la aplicación (1).
    - Registrada y revisada (3).
    - Registrada pero no revisada (6).
    - No registrada (9).

3.6. Factores para estimar el impacto

Es importante tener en cuenta que existen dos tipos de impactos. El primero es el “impacto técnico” en la aplicación, los datos que utiliza, y las funciones que proporciona. El otro es el “impacto sobre negocio”, sobre el negocio y la compañía que opera la aplicación. Al final, el impacto sobre el negocio es más importante. Sin embargo, es posible que no se tenga acceso a toda la información necesaria para identificar las consecuencias provocadas por la explotación exitosa de una vulnerabilidad. En este caso, se debe detallar todo sobre el riesgo técnico que permita al representante del negocio tomar una decisión sobre el riesgo.

3.6.1. Factores de impacto técnico

El impacto técnico se puede dividir en factores alineados con las tradicionales áreas de seguridad: confidencialidad, integridad, disponibilidad y control de responsabilidad. El objetivo es estimar la magnitud del impacto en el sistema si la vulnerabilidad es explotada.

Algunos factores serían la pérdida o corrupción de los datos, falta de integridad, o negación de acceso. Podrían permitir a los atacantes ejecutar secuencias de comandos en el navegador de una víctima para secuestrar las sesiones de usuario, destruir sitios web, insertar código hostil, redirigir usuarios,

instalar código malicioso en el navegador de la víctima. Los atacantes pueden obtener acceso no autorizado a datos o funcionalidad del sistema. Cualquiera de estos factores resulta un riesgo para todo el sistema si una cuenta de administración es comprometida. Se podría exponer información asociada a los usuarios y pueden derivar en un robo de cuentas. A continuación se mencionan algunos de los factores de impacto técnico.

- **Pérdida de confidencialidad:** Cantidad y sensibilidad de la información que podría ser revelada y cuán delicada es.
    - Revelación mínima de datos no sensibles (8).
    - Revelación mínima de datos críticos (6).
    - Amplia revelación de datos no sensibles (6).
    - Amplia revelación de datos críticos.
    - Todos los datos revelados (1).
- **Pérdida de integridad:** Cantidad de datos se podrían corromper y el daño que sufre.
    - Mínimo, datos ligeramente corruptos (9).
    - Mínimos datos seriamente dañados (3).
    - Gran cantidad de datos ligeramente dañados (5).
    - Todos los datos totalmente corruptos (1).
- **Pérdida de disponibilidad:** Servicios que se pueden ver interrumpidos y su vitalidad.
    - Mínimo número de servicios secundarios interrumpidos (9).
    - Mínimo número de servicios primarios interrumpidos (5).

  - Gran número de servicios secundarios interrumpidos (5).
  - Gran número de servicios primarios interrumpidos (2).
  - Todos los servicios perdidos (1).

- **Pérdida de control de responsabilidad:** Posibilidad de trazar las acciones de los atacantes hasta llegar a un individuo.
  - Totalmente trazable (9).
  - Es posible que se pueda trazar (2).
  - Completamente anónimo (1).

3.6.2. Factores de impacto sobre el negocio

El impacto sobre el negocio proviene del impacto técnico, pero requiere un conocimiento profundo sobre que es importante para la compañía que utiliza la aplicación. En general, deberías considerar los riesgos teniendo en cuenta el impacto sobre el negocio, particularmente si tu audiencia es del nivel ejecutivo. El riesgo sobre el negocio es lo que justifica la inversión en solucionar problemas de seguridad.

Muchas compañías tienen una guía de clasificación de activos y/o una referencia del impacto sobre el negocio para ayudar a formalizar qué es importante para su negocio. Estos estándares pueden ayudar a centrarte en las cuestiones de seguridad verdaderamente importantes. En caso de no estar disponibles, se habla con las personas que comprenden el negocio para obtener su punto de vista acerca de que es importante.

El sistema puede estar en riesgo sin que se pueda tener conocimiento de este hecho y los datos pueden ser robados o modificados, además los costos de recuperación pueden ser altos. Se debe tener en cuenta el valor de la

información, datos o funciones expuestos en los canales de comunicación en cuanto a sus necesidades de confidencialidad e integridad. Muy importante se debe considerar el impacto que tiene en la reputación del negocio y en la exposición pública de la vulnerabilidad.

Los factores que se exponen debajo son áreas comunes a muchos negocios, pero esta área es incluso más particular para una compañía que los factores relacionados con el agente que provoca la amenaza, la vulnerabilidad, y el impacto técnico.

- **Daño Financiero:** Daño financiero resultado de la explotación de una vulnerabilidad.
    - Menor al coste de arreglar la vulnerabilidad (1).
    - Leve efecto en el beneficio anual (5).
    - Efecto significativo en el beneficio anual (8).
    - Bancarrota (9).
- **Daño sobre la reputación:** La explotación de una vulnerabilidad tendría por resultado un daño sobre la reputación.
    - Daño mínimo (1).
    - Pérdida de las cuentas principales (4).
    - Pérdida del buen nombre (5).
    - Daño sobre la marca (9).
- **No conformidad:** Exposición introduce la no conformidad.
    - Violación leve (2)
    - Clara violación (5)

- Violación prominente (8)

- **Violación de la privacidad:** Cantidad de información que facilite la identificación personal podría ser revelada.
  - Un individuo (1)
  - Cientos de personas (5)
  - Miles de personas (7)
  - Millones de personas (9)

3.6.3. Determinación de la gravedad del riesgo

Para determinar la probabilidad de ocurrencia estimada y el impacto estimado se debe calcular la severidad global del riesgo. En este caso lo que identifica es si la probabilidad de ocurrencia es baja, media o alta y después se debe hacer lo mismo con el impacto.

Se necesita defender las puntuaciones o hacerlas repetibles, de esta forma se podría seguir un proceso más formal para puntuar los factores y calcular el resultado. Hay que tener en cuenta que en estas estimaciones existe bastante incertidumbre, y que esos factores tienen por objetivo ayudar a alcanzar un resultado razonable. Primeramente se selecciona una de las opciones asociadas con cada factor y se introduce el número asociado en la tabla. Luego se toma la media de las puntuaciones y se calcula la probabilidad de ocurrencia global.

La Tabla 2 muestra una prueba con valores para probar los datos asociados al agente causante de la amenaza y a los factores asociados a la vulnerabilidad. Se muestra el riesgo Almacenamiento Criptográfico Inseguro en el sistema SAPLA. Cada aspecto tiene asociado un número que significa la

probabilidad de ocurrencia y niveles de impacto. Después suman y se dividen entre el total para determinar la probabilidad de ocurrencia global. En este caso el resultado es 7.125, lo que significa que la probabilidad es alta.

Tabla 2: Probabilidad de ocurrencia. Fuente: (OWASP, 2014).

Factores correspondientes al agente causante de amenaza	Nivel de habilidad	6
	Motivo	9
	Oportunidad	7
	Tamaño	5
Factores asociados a la vulnerabilidad	Factibilidad descubrimiento	9
	Factibilidad explotación	5
	Concienciación	9
	Detección de intrusión	8
Probabilidad de Ocurrencia Global= 7.125 (ALTA)		

Además se debe conocer el impacto técnico global y el impacto global sobre el negocio, ambos procesos son similares al anterior. En estos casos la respuesta se puede deducir fácilmente, cuando la mayoría de los valores son altos, medios o bajos. Aunque la respuesta sea obvia es recomendable realizar la estimación basada en los factores. Primero se calcula la media de las puntuaciones para cada uno de los factores, igual que el caso anterior menos que 3 se considera bajo, de 3 a 6 medio y de 6 a 9 altos. Para cada una de las estimaciones de los factores en la Tabla 3 se muestra una prueba con sus respectivos valores. En este caso los números se suman y se dividen entre la cantidad de aspectos que se evaluaron en cada uno de ellos. En la Tabla 3 se muestran los datos asociados al riesgo Almacenamiento Criptográfico Inseguro en el sistema SAPLA. Para este caso el Impacto técnico global estimado es de 7.25 por lo cual se considera alto y el Impacto global sobre el negocio es de

7.25 lo que significa que es alto. Con estos resultados la entidad debe definir qué estrategia seguir de acuerdo a sus intereses.

Tabla 3: Impacto de ataque. Fuente: (OWASP, 2014)

Impacto Técnico	Perdida de confidencialidad	9
	Perdida de Integridad	7
	Perdida de disponibilidad	5
	Pérdida de control responsabilidad	8
Impacto Técnico Global = 7.25 (ALTO)		
Impacto sobre Negocio	Daño Financiero	9
	Daño a la Reputación	9
	No conformidad	5
	Violación de Privacidad	6
Impacto Global sobre el Negocio = 2.25 (BAJO)		

La Tabla 4 muestra una prueba con valores para probar los datos asociados al agente causante de la amenaza y a los factores asociados a la vulnerabilidad. En la siguiente tabla se muestran los datos asociados al riesgo Protección insuficiente en la Capa de Transporte en el sistema SOCI. El procedimiento se realiza igual que en el caso anterior. El resultado es 6.375, lo que significa que la probabilidad de que se manifieste este riesgo en la aplicación es media.

Tabla 4: Probabilidad de ocurrencia. Fuente: (OWASP, 2014).

Factores correspondientes al agente causante de la amenaza	Nivel de Conocimiento	4
	Motivación	7
	Oportunidad	8
	Tamaño	6
Factores asociados a la Vulnerabilidad	Facilidad de descubrimiento	7
	Facilidad de explotación	4
	Conocimiento	7
	Detección de Intrusión	8
Probabilidad de ocurrencia global=6.375 (MEDIA)		

La Tabla 5 muestra los datos asociados al riesgo Protección insuficiente en la Capa de Transporte en el sistema SOCI. Para este caso el Impacto técnico global estimado es de 6.75 por lo cual se considera medio y el Impacto global sobre el negocio es de 5.75 lo que significa que es medio. Con estos resultados la entidad debe definir qué estrategia seguir de acuerdo a sus intereses.

Tabla 5: Impacto del ataque. Fuente: (OWASP, 2014)

Impacto Técnico	Perdida de confidencialidad	8
	Perdida de Integridad	7
	Perdida de disponibilidad	4
	Pérdida de control responsabilidad	8
Impacto Técnico Global = 6.75 (MEDIO)		
Impacto sobre Negocio	Daño Financiero	7
	Daño a la Reputación	7
	No conformidad	5
	Violación de Privacidad	4
Impacto Global sobre el Negocio = 5.75 (MEDIO)		

Después del análisis realizado se percibe a través de las tablas anteriores que el riesgo Almacenamiento Criptográfico Inseguro posee mayor probabilidad de ocurrencia que el riesgo Protección insuficiente en la Capa de Transporte. Además de que el impacto técnico global y el impacto global sobre el negocio son superiores.

3.6.4. Determinando la severidad

Aunque se hayan determinado los impactos estimados y la probabilidad de ocurrencia, es necesario fusionar ambos resultados para obtener la calificación final de la severidad global. Esta puntuación final depende en gran medida de la información que se conozca sobre el impacto en el negocio o el impacto técnico. Se debe emplear el impacto sobre el cual se tenga mayor información,

para poder determinar una aproximación real de la severidad global del riesgo en la empresa. Es decir si se dispone de mayor información sobre el impacto que tiene el riesgo sobre el negocio, entonces este es el que se debe emplear. De lo contrario si se dispone de mayor información sobre el impacto técnico, entonces este es el que se emplea. En caso de que se tenga buena información sobre los dos impactos, se tendrán en cuenta ambos para determinar la calificación final de la severidad global.

Es necesario que la empresa comprenda el contexto de las vulnerabilidades que se evalúan, para poder determinar si es o no alto y ver si se pueden o no tomar buenas decisiones respecto al riesgo. Se debe tener cuidado en la comprensión de este contexto, porque una mala interpretación puede conllevar a la falta de confianza entre los equipos de negocio y de seguridad que están presentes en las organizaciones.

3.6.5. Decidir qué se debe corregir

Después de haber clasificado los riesgos de cualquier aplicación, se debe realizar una lista priorizada de qué arreglar. Esta lista se elabora por la empresa a partir de sus necesidades y prioridades principales. Inicialmente, se deben arreglar los problemas que supongan un riesgo más severo. Resolver primero los riesgos menos importantes no ayuda a reducir el riesgo global, incluso si la solución es fácil o de bajo costo. Después se deben arreglar los riesgos que no son tan importantes y que su probabilidad e impacto en el negocio y técnico no son altos. Por último se arreglan los menos significativos, pues realmente son importantes los que afectan severamente el sistema, por eso inicialmente todas las personas involucradas se deben enfocar en

resolverlos. En el caso de los sistemas analizados se debe corregir: (OWASP Testing Guide v2.0, 2007)

- En los sistemas SAPLA y SPI se deben utilizar algoritmos de encriptación de datos para proteger la información. Se recomienda utilizar algoritmos asimétricos que son más factibles en cuanto a seguridad.
- En el caso de SOCI se debe realizar un estudio para ver como sustituir el algoritmo que se utiliza actualmente MD5 por otro más fuerte.
- Las tres aplicaciones deben utilizar algún protocolo criptográfico, que proporcione seguridad en la autenticación y privacidad de la información, como por ejemplo SSL Y TLS.

3.6.6. Personalización del modelo de calificación de riesgos

Disponer de un marco de puntuación del riesgo que sea personalizable para el negocio es una cuestión crítica que se debe adoptar. Un modelo a medida es más probable que produzca resultados que concuerden con las percepciones de la gente sobre lo que es un riesgo serio. Existen varios modos de adaptar este modelo a una organización.

- **Añadiendo factores:** Se pueden escoger diferentes factores que representen lo que es importante para la organización. Por ejemplo, en los sistemas SAPLA y SPI se pueden añadir factores de impacto relacionados con la inseguridad de los datos almacenados pues no se encriptar. También pueden añadir factores de probabilidad de

ocurrencia, como la oportunidad de un atacante o la fortaleza de un algoritmo de codificación.

- **Personalizando opciones:** Existen opciones de ejemplo asociadas a cada factor, pero el modelo es más efectivo si se personalizan las opciones para el negocio. Por ejemplo, utilizar nombres de los diferentes grupos y los propios nombres para diferentes clasificaciones de la información. También se puede cambiar las puntuaciones asociadas con las opciones. El mejor modo de identificar las puntuaciones correctas es comparar las puntuaciones producidas por el modelo con las producidas por un equipo de expertos. Por lo tanto, se puede afinar el modelo ajustando cuidadosamente las puntuaciones.
- **Ponderando factores:** El modelo anterior asume que todos los factores son igualmente importantes. Se puede asignar un peso a los factores para enfatizar aquellos que son más significativos para el negocio. Esto permite utilizar una media ponderada, logrando que el modelo sea un poco más complejo. Pero por lo demás todo funciona similar, asimismo se puede afinar el modelo contrastándolo con puntuaciones del riesgo que se consideren precisas. (OWASP Testing Guide v2.0, 2007).

3.7. Evaluación comparativa de los riesgos identificados

A continuación se muestra una tabla donde se comparan los 2 riesgos identificados en base al Top 10 de OWASP. La tabla 6 incluye tres factores de probabilidad para cada debilidad (prevalencia, detectabilidad y facilidad de explotación), dos factores de impacto (impacto técnico e impacto en el negocio)

y los vectores de ataques. A continuación se describe cada uno de los elementos presentes en la Tabla 6:

- Agentes de amenazas: son los que realizan el ataque a la aplicación.
- Vectores de ataques: son las diferentes formas en las que se puede llevar a cabo el ataque. Pueden ser cadenas de texto que explotan la sintaxis del intérprete atacado.
- Prevalencia de debilidades: es la frecuencia con la que se manifiesta el riesgo analizado.
- Detectabilidad: es la probabilidad o facilidad con que se puede detectar un riesgo.
- Impacto técnico: es el la magnitud del daño que ocasiona el riesgo a la aplicación.
- Impacto en el negocio: es el daño que sufre el negocio con el éxito del riesgo. (OWASP.org, 2014)

Tabla 6: Riesgos

	Riesgo 7	**Riesgo 9**
Agentes de Amenaza	Usuarios y administradores del sistema.	Personas que puedan capturar el tráfico de red de sus usuarios.
Vectores de Ataques	Difícil	Difícil
Prevalencia de Debilidades	Poco Común	Común
Defectibilidad de Debilidades	Difícil	Fácil
Impacto Técnico	Severo	Moderado
Impacto en el Negocio	Severo	Moderado

A cada uno de los aspectos se le asigna una clasificación, en el caso de los vectores de ataques y la detectabilidad de debilidades se clasifican en Difícil,

Medio o Fácil, estos niveles se definen en cuanto a la complejidad de los ataques o capacidad de detección. Otros aspectos importantes son el impacto técnico y el impacto en el negocio, los cuales se clasifican en Severo o Moderado. Por último y no por eso menos importante, está la prevalencia de debilidades la cual se clasifica en Muy Difundida, Muy Común, Común o Poco Común, según la probabilidad de ocurrencia.

Después de realizar la comparación se concluye que la presencia del Riesgo 7: Almacenamiento Criptográfico Inseguro es más perjudicial para la entidad, ya que con la ocurrencia de este riesgo se comprometen todos los datos que deberían haber estado cifrados, tales como cuentas de usuario y datos personales. Además puede causar grandes pérdidas cualitativas y cuantitativas al negocio. También afecta a la aplicación pues se pueden mostrar públicamente las vulnerabilidades que presenta.

Se concluye además que el Riesgo 9: Protección insuficiente en la Capa de Transporte es también perjudicial para la empresa pues tiene un impacto moderado sobre el negocio. Es necesario evitar que la información esté expuesta en los canales de comunicación, en cuanto a sus necesidades de confidencialidad e integridad. Además si una cuenta de administración es comprometida, puede verse expuesta toda la aplicación.

Después de analizar ambos riesgos se recomienda a la empresa que debe enfocarse en resolver el Riesgo 7 primero, pues tanto el impacto técnico como el impacto sobre el negocio se afectan severamente cuando está presente en las aplicaciones. Se recomienda que el código de cada una de las aplicaciones que se desarrolle deba ser revisado contra estas vulnerabilidades, pues estas

fallas están siendo utilizadas frecuentemente por los atacantes. Los proyectos que estén en desarrollo deben considerar estas vulnerabilidades en la documentación que realizan en cada una de las fases de requerimientos, diseño, implementación y pruebas. Deben asegurarse que en las aplicaciones no han sido introducidas estas vulnerabilidades y en caso de que existan deben ser eliminadas de forma correcta. La empresa debe destinar presupuesto y tiempo para estas actividades de seguridad en sus aplicaciones, para lograr crear productos de desarrollo de software con la calidad necesaria. Se deben implementar políticas de desarrollo seguro, mecanismos de diseño, pruebas de intrusión y revisión de seguridad al código fuente.

En la actualidad la seguridad en las aplicaciones ya no es una opción, es una necesidad. Cada vez más aumentan los ataques, por lo tanto las organizaciones deben establecer un mecanismo eficaz que permita asegurar las aplicaciones. Dada la cantidad de soluciones que se realizan diariamente para satisfacer las solicitudes que existen por parte de los clientes, muchas organizaciones luchan por gestionar el enorme volumen de vulnerabilidades. OWASP recomienda a las organizaciones establecer un programa de seguridad para aumentar el conocimiento y mejorar la seguridad de las aplicaciones. Para conseguir un nivel avanzado de seguridad, se requiere que todas las partes de la empresa trabajen en conjunto y de manera eficiente. Se requiere que la seguridad sea visible, para que todos los que participen entiendan y estén conscientes de la postura de la entidad en cuanto a la seguridad en las aplicaciones que realizan. También es necesario centrarse en

las actividades y resultados que ayudan realmente a mejorar la seguridad de la empresa mediante la reducción de los riesgos.

CAPÍTULO 4

PROPUESTA DE BUENAS PRÁCTICAS DE SEGURIDAD

Para minimizar los daños que puedan sufrir la información o la infraestructura informática la seguridad informática establece restricciones. Las mismas definen por lo general protocolos, horarios de funcionamiento, denegaciones, planes de emergencia y, perfiles de usuario, autorizaciones para garantizar un alto nivel de seguridad informática en las organizaciones requeridas; minimizando así el impacto de los riesgos en el desempeño de los trabajadores y de la organización.

Para proteger activos informáticos como la infraestructura, los usuarios y la información fue creada la seguridad informática. La infraestructura es fundamental para el funcionamiento de la organización, la gestión y el almacenamiento de la información. La función de la seguridad informática en esta área es velar que los equipos funcionen adecuadamente y anticiparse en caso de fallas. Los usuarios son las personas que utilizan la estructura tecnológica, zona de comunicaciones y que gestionan la información. Debe protegerse el sistema en general para que el uso por parte de ellos no pueda poner en peligro la seguridad de la información. Además se debe evitar que la información que manejan o almacenan sea vulnerable. La información es el principal activo. Utiliza y reside en la infraestructura computacional y es utilizada por los usuarios (Escobar, 2012).

4.1. Buenas prácticas

Las buenas prácticas hacen alusión a sistemas de calidad que establecen las condiciones bajo las que se planifican, elaboran, controlan, registran y

archivan los datos obtenidos, con el fin de asegurar la fiabilidad de los mismos. (Cabré, 2009) Seguros de su incidencia, efecto e impacto positivo, se difunden las experiencias realizadas y el conocimiento adquirido, para poder transferirlas y hacerlas replicables en otros contextos (González Ramírez, 2007)

Criterios de selección

Una buena práctica cuenta con un mínimo de criterios esenciales, los cuales se detallan a continuación:

- **Documentada:** para servir de referente a otros y facilitar la mejora de sus procesos. Este es un sentido esencial de la buena práctica, por lo tanto debe de estar documentada, de modo que pueda trasladarse el conocimiento fácilmente a otra organización para aprender a realizarla.
- **Accesible:** para ser utilizada desde cualquier lugar y por cualquier persona.
- **Basada en procesos y metodologías:** una buena práctica cuenta con metodologías que han sido seleccionadas cuidadosamente para transformar la situación priorizada
- **Probada e implementada:** las buenas prácticas se encuentran probadas o en proceso y en el mejor de los casos han sido o están siendo evaluadas.
- **Capaz de establecer objetivos:** las buenas prácticas responden a una necesidad identificada, son fruto de una evaluación cuidadosa de alguna

característica en una población definida que se hace necesario modificar y mejorar y por tanto tiene objetivo definido, relevante y realista.

- **Transferible:** como finalidad deben promover la transferencia de conocimiento sobre los métodos, las herramientas y los enfoques utilizados para la puesta en marcha de aquellas experiencias o iniciativas consideradas como Buenas Prácticas.
- **Sostenible:** el ingreso excede el costo. La relación entre ingreso y coste es mejor que la de prácticas similares.
- **Eficiente:** la relación entre ingreso y coste es mejor que la de prácticas similares.
- **Eficaz:** conduce a los resultados esperados.

Estas características representan los principales criterios de selección de las buenas prácticas.

4.2. Buenas prácticas de seguridad para el desarrollo web

Son el conjunto de elementos o acciones que se llevan a cabo para garantizar la seguridad en las aplicaciones web desde temprano en su desarrollo hasta su finalización y mantenimiento.

Para implementar buenas prácticas de seguridad en la web se deben tener en cuenta varios elementos que se describen a continuación.

Garantizar la correcta entrada de datos por parte de los usuarios finales es una medida importante, las cuales deben estar listas para capturar excepciones y brindar una respuesta controlada. Para ello se deben realizar validaciones tanto del lado del cliente utilizando JavaScript como del lado del servidor por

medio de rutinas del lenguaje de programación. Siendo ambas muy importantes porque las validaciones del lado del cliente pueden pasarse por alto o desactivarse. Se implementa filtrando datos proporcionados por el usuario del lado del servidor. Es recomendable el uso de listas blancas, las cuales pueden generarse a partir del empleo de expresiones regulares que evitan asignar directamente el valor obtenido de un formulario a una variable, garantizando que los datos se validen correctamente. Se deben realizar validaciones correctamente porque de lo contrario pueden ingresarse cadenas de código malicioso para obtener información de la aplicación o de sus usuarios, con lo cual se corre el riesgo de SQL Injection, XSS y/o CSRF.

Se debe tener en cuenta la implementación de sesiones la cual permite: el seguimiento al usuario, mantener valores de variables a través del sitio sin tener que emplear campos ocultos en formularios y restringir el acceso a determinados elementos. Es muy importante dar seguimiento a la sesión, así como iniciarla y cerrarla de forma correcta, para evitar violaciones de seguridad. Como recomendación se debe establecer: el uso de sesiones donde el usuario comience la interacción "restringida" con la aplicación; que la sesión continúe activa en cada elemento visitado para su visualización en caso contrario se debe redirigir al usuario a iniciar la sesión; y el correcto cierre de la sesión cuando el usuario termine de visitar el sitio. Los riesgos de una implementación inadecuada se encuentran relacionados con: acceso a recursos restringidos, robo de identidad de otro usuario y mal uso de los recursos de la aplicación.

La gestión de los datos de los usuarios finales se debe realizar con protección adicional lo primero a valorar es el cifrado del canal de comunicación para el intercambio entre el cliente y el servidor. Es muy importante mantener seguros los datos proporcionados por los usuarios, tanto al inicio de sesión (identificador, contraseña) como al registrarse en formularios (nombre, apellidos, dirección, correo electrónico, teléfono). Se recomienda para ello establecer el uso del protocolo HTTPS mediante la instalación de un certificado en el servidor de aplicaciones web y configurar el servidor para su empleo ya sea en el sitio principal o por cada virtual host. Como riesgo puede ocurrir la intercepción del tráfico de red, mostrando información sensible en tránsito (man in the middle).

En la web al obtener un desarrollo a la medida se pueden configurar los mensajes de error que se muestran a los usuarios para evitar que se observe información del sitio. Por tanto deben considerarse los fallos que puedan ocurrir en la aplicación y la información mostrada a los usuarios, como su nombre lo indica fallo seguro. Se recomienda el uso de mensajes genéricos de error de ser necesario. Como riesgo de implementación inadecuada podría ocasionar la exposición accidental de información de instalación y configuración del sitio (versiones, software empleado, rutas del sistema).

La información es el activo más importante manejado por las aplicaciones web y el que más vulnerable nos hace en materia de seguridad informática, por lo cual se debe proteger bien. Para ello es recomendable tomar las medidas necesarias como: disminuir el acceso de personas no deseadas a la información que se maneja en la aplicación, configurar el servicio de bases de

datos de forma segura haciendo hardening a la configuración y evitar usuarios, contraseñas y configuraciones por defecto. Como riesgo de implementación inadecuada puede ocurrir la extracción o modificación de información de la base de datos (del servicio y del contenido).

En ocasiones es necesario que la aplicación emplee funcionalidad o incluya contenido de terceros (módulo de validación, calendarios, autenticación con oauth, conexión con twitter o facebook), si así fuere como prevención se debe investigar sobre las vulnerabilidades existentes para dicho contenido. Es muy importante emplear, sólo en caso necesario la mínima funcionalidad correspondiente y no instalar contenido o funcionalidad que no se vaya a emplear. En el caso de instalar bibliotecas, módulos o funcionalidad extra por parte de terceros, es conveniente instalar la última versión estable y que cuente con las actualizaciones de seguridad correspondientes. Los riesgos dependerán del tipo de vulnerabilidad del contenido, pero puede incluir XSS, CSRF, SQL Injection, robo de sesión, escalada de privilegios, entre otros. (Aguilar & Hernández, 2013)

4.3. Estrategia

La estrategia de seguridad permite a los administradores de seguridad proteger la disponibilidad, integridad y confidencialidad de los datos de los sistemas informáticos de las organizaciones. La estrategia se debe poner en práctica sistemáticamente como precaución, e implicar de planes de contingencia en caso de situaciones excepcionales.

Los administradores deben decidir el tiempo, el capital y el esfuerzo a invertir para desarrollar las directivas y controles de seguridad apropiados.

Cada organización debe analizar sus necesidades específicas y determinar sus requisitos y limitaciones en cuanto a recursos y a programación. Cada sistema informático, entorno y directiva organizativa es diferente, lo cual provoca que cada servicio y estrategia de seguridad sean exclusivos. A continuación se detallan principios a tener en cuenta para obtener una buena estrategia de seguridad:

a. **Actualización constante de los métodos herramientas y técnicas de seguridad informática**

   Las organizaciones disponen de listas de amenazas que ayudan a los administradores de seguridad a identificar los diferentes métodos, herramientas y técnicas de ataque más probables. Se hace necesario que esta área de conocimiento se encuentre en constante actualización para evitar así desconocimientos, vulnerabilidades y violaciones de seguridad.

b. **Definición de estrategias preventivas y de contingencias**

   Los planes de seguridad en cada organización deben incluir estrategias de prevención de ataques, las cuales son un conjunto de pasos que reducen al mínimo la cantidad de puntos vulnerables existentes en las directivas de seguridad. Evaluar constantemente las debilidades contribuye a desarrollar la estrategia preventiva. La estrategia de contingencia es la que se realiza posterior al ataque donde se evalúan los daños que ha causado el ataque, se repara, se documenta, se aprende de la experiencia.

c. Emular ataques en entornos de pruebas

Realizar ataques simulados en entornos de pruebas o en laboratorios garantiza evaluar los lugares donde hay puntos vulnerables ajustando las directivas y los controles de seguridad en consecuencia. Estas pruebas no se deben llevar a cabo en los sistemas real, pues el resultado puede ser desastroso, pero son de vital importancia aplicarlas debido a los riesgos y consecuencias de los ataques.

d. Control de incidencias

Se recomienda establecer un equipo de control a incidentes. Este equipo debe estar implicado en los trabajos preventivos de seguridad. Deben definir instrucciones, herramientas, investigaciones, y ejecutar tareas para controlar incidentes relativos a ataques al sistema.

4.4. Generación de buenas prácticas en el desarrollo de software

Luego del análisis realizado a los sistemas: Sistema de Población de Identificaciones (SPI), Sistema de Auditoría de Prevención de Lavado de Activos (SAPLA), Sistema para otorgar Credenciales a Intermediarios de Seguros (SOCI) de la SBS se determinaron que existe mayor probabilidad de incurrir en los siguientes riesgos debido: a las características que presentan los mismos, al objeto social al cual están destinados y a las especificaciones técnicas particulares de cada uno:

- Riesgo 7: Almacenamiento Criptográfico Inseguro.
- Riesgo 9: Protección insuficiente en la capa de transporte.

En base a los riesgos listados anteriormente se describe a continuación buenas prácticas que se deben tomar en cuenta en el desarrollo y

mantenimiento de los sitios de la SBS, para lograr así aplicaciones que garanticen la navegación, la confidencialidad y la integridad de los datos y de la información mostrada.

Para el Riesgo 7: Almacenamiento Criptográfico Inseguro, algunas aplicaciones web no protegen adecuadamente los datos sensibles, credenciales de autenticación con mecanismos de cifrado o hashing. Atacantes pueden modificar o robar tales datos protegidos inadecuadamente para conducir robos de identidad, fraudes de tarjeta de crédito u otros crímenes.

Las Buenas Prácticas asociadas el Riesgo 1. Almacenamiento Criptográfico Inseguro primeramente se deben identificar los datos que son suficientemente sensibles y requieren cifrado. Por ejemplo, contraseñas, información personal de entidades controladas y para ellos se deben tener en cuenta:

- Valorar todos los riesgos que puedan afectar a los datos teniendo en cuenta los ataques internos y los usuarios externos.
- Garantizar datos sensibles cifrados en todos aquellos lugares donde son almacenados durante períodos largos.
- Comprobar el cifrado de las copias de seguridad que se almacenan externamente.
- Verificar que las contraseñas se gestionan y almacenan por separado.
- Permitir acceso solo a usuarios autorizados a los datos no cifrados.
- Utilizar un algoritmo estándar seguro.
- Poseer claves seguras, protegidas ante accesos no autorizados.

- Garantizar el respaldo de claves mediante un algoritmo robusto como un hash.
- Elaborar un plan para el cambio de contraseñas.

Para el Riesgo 7: Protección insuficiente en la capa de transporte, las aplicaciones frecuentemente fallan al autenticar, cifrar, y proteger la confidencialidad e integridad de tráfico de red sensible. Cuando esto ocurre, es debido a la utilización de algoritmos débiles, certificados expirados, inválidos, o sencillamente no utilizados correctamente.

Las Buenas Prácticas asociadas el Riesgo 9. Protección insuficiente en la capa de transporte se debe proporcionar una protección adecuada a la capa de transporte puede afectar al diseño de la aplicación. De esta forma, resulta más fácil requerir SSL para la aplicación completa. Por razones de rendimiento, algunas aplicaciones utilizan SSL únicamente para acceder a páginas privadas. Otras, utilizan SSL sólo en páginas “críticas”, pero esto puede exponer identificadores de sesión y otra información sensible. Se debe aplicar lo siguiente:

- Utilizar SSL en las páginas más críticas y redireccionar las peticiones sin SSL a las páginas que si lo tienen.
- Establecer el atributo “secure” en todas las cookies de las páginas con mayor riesgo.
- Configurar el servidor SSL para que acepte únicamente algoritmos considerados fuertes.

- Verificar que el certificado sea válido, no se encuentre expirado o revocado y que se ajuste a todos los dominios utilizados por la aplicación.
- Conexiones a sistemas finales (back-end) y otros sistemas también deben utilizar SSL u otras tecnologías de cifrado.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Basado en que las acciones que realizan son claves para el negocio, que el manejo de información es elevado, por ser sistemas que constantemente actualizan sus datos y son esenciales para el correcto funcionamiento de SBS, se seleccionaron algunas de las aplicaciones de la SBS que se encuentran desarrolladas bajo la plataforma JEE; fueron seleccionadas para la presente investigación el Sistema de Población de Identificaciones (SPI), Sistema de Auditoría de Prevención de Lavado de Activos (SAPLA), Sistema para otorgar Credenciales a Intermediarios de Seguros (SOCI).
- Fueron identificadas y discutidas las vulnerabilidades, amenazas y riesgos más comunes presentes en una aplicación Web, a partir del estudio realizado sobre las tecnologías web, las metodologías aplicadas para el desarrollo sobre plataformas JEE, las tecnologías utilizadas y las arquitecturas.
- Se realizó una evaluación de los riesgos basados en el Top Ten de OWASP, de las vulnerabilidades, amenazas, el impacto, detectándose que en las aplicaciones de la SBS hay dos riesgos bien claros que presentan una vulnerabilidad media-alta, al igual que la amenaza, pudiendo llegar el impacto a ser grande en caso de ser explotada; estos

riesgos son: Almacenamiento Criptográfico Inseguro y Protección Insuficiente en la Capa de Transporte.

- Se realizó una lista de buenas prácticas para asegurar las aplicaciones, corregir los riesgos detectados y los que estén por surgir en el proceso de desarrollo de nuevas funcionalidades en las aplicaciones estudiadas, partiendo de los resultados obtenidos y siguiendo los resultados encontrados en la investigación.

5.2. Recomendaciones

- Se recomienda aplicar de inmediato la recomendación para eliminar el Riesgo 1, debido al elevado impacto negativo que tendría en las aplicaciones en caso de ser explotado.
- Desarrollar una guía estandarizada de la información a recoger o brindar por las empresas o clientes cuando necesiten que sus sistemas sean analizados según OWASP.
- Se propone en futuros trabajos automatizar el proceso de estudio de vulnerabilidades en aplicaciones web, basado en las especificaciones obtenidas de la recomendación anterior.
- Analizar según las actualizaciones OWASP 2013, qué nuevos aspectos deben ser tenidos en cuenta para futuros trabajos.

BIBLIOGRAFÍA Y WEBGRAFÍA

OWASP Testing Guide v2.0. (2007).

Web Seguro. (2012). BuenasTareas.com.

Aghaei, S., Nematbakhsh, M., & Khosravi Farsani, H. (2012). Evolution of the World Wide Web from web 1.0 to web 4.0. International Journal of Web & Semantic Technology (IJWesT).

Aguilar, A., & Hernández, A. (2013). Sugerencias de Seguridad para Sitios Web.

Areitio, J. (2008). Seguridad de la Información. Redes, informática y sistemas de información. Cengage Learning Paraninfo, S.A.

Aumaille, B. (2002). J2EE Desarrollo de Aplicaciones Web. Ediciones ENI.

Beust, C. D. (2002). Programación Java Server con J2EE (1.3 ed.). España: Anaya Multimedia.

Bupartech. (2013). Manual de uso de la Herramienta de Apoyo al Manual Único de Supervisión (HAMUS). QUito: Superintendencia de Bancos y Seguros del Ecuador.

Cabré, C. (2009). Terminología y buenas prácticas". Actas de la Conferencia 2009 (Publifarum, nº 12, 2010 ed.).

Donate, F. P. (2005). Transmisión de imágenes de video mediante Servicios Web XML sobre J2ME. Universidad de Sevilla. . Recuperado el 2014, de http://bibing.us.es/proyectos/abreproy/11372/fichero/

ERODATA. (2012). Superintendencia de Bancos y Seguros del Ecuador. Manual de uso del Sistema para otorgar Credenciales a Intermediarios de Seguros (SOCI).

Erodata, S. (2012a). Manual Técnico del Sistema para otorgar Credenciales a Intermediarios de Seguros (SOCI).

Escobar, L. (2012). Diseño de un Sistema de información para la optimización del uso de la intranet en la empresa imágenes gráficas S.A. Santiago de Cali.

Fuchs, & et al. (2010). Theoretical Foundations of the Web: Cognition, Communication, and Co-Operation. Towards an Understanding of Web 1.0, 2.0, 3.0. Future Internet.

González Ramírez, T. (2007). El concepto de `buenas prácticas´: origen y desarrollo (Vol. nº 222).

Gosling, J. (2002). The Java Language Specification (Second Edition ed.). Sun Microsystems,Inc.

Hendler, J. (2009). Web 3.0 Emerging. Computer (Vol. 42(1)).

ISO. (2013). ISO 27000 Directory . Recuperado el 2014, de http://www.27000.org/

Jonhson, R. (2003). Expert One-on-One J2EE Desing and Development Wrox Press.

López Trujillo, M., Marulanda Echeverry, C. E., & Vega, O. A. (2012). Servicios de Gestión de Conocimiento Utilizando La Computación en Nube.

Mateu, C. (2004). Desarrollo de Aplicaciones Web. F. p. a. l. U. O. d. Catalunya Ed.Eureca Media, SL.

Mateu, C. (2004). Desarrollo de Aplicaciones Web. Cataluña: Eureca Media S.L.

Microsoft. (2003). Improving Web Application Security.Threats and Countermeasures.

O'Reilly, T. (2007). What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software (Vol. 1). Communications & Strategies.

OWASP. (2012).

OWASP. (2014 de Mayo de 2014). OWASP Risk Rating Methodology. Recuperado el 26 de Diciembre de 2013, de https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

OWASP. (29 de 3 de 2014). OWASP Top Ten 2010 Proyect. Recuperado el 2 de 04 de 2014, de https://www.owasp.org/images/2/2d/OWASP_Top_10_-_2010_FINAL_(spanish).pdf

OWASP.org. (2014). The Open Web Application Security Project. Recuperado el 2014

Palmer, S. (2011). Web Application Vulnerabilities: Detect, Exploit, Prevent. (I. Elsevier, Ed.) Syngress Publishing, Inc.

Paredes, B. (2014). Especificaciones Técnicas del Sistema de Auditorias de Prevención de Lavado de Activos (SAPLA).

Parravicini, L. (2011). Programación web segura.

Pérez Herrera, E. (2003). Tecnologías y redes de transmisión de datos. México: Limusa.

Pickin, & et al. (2010). Introducción a las arquitecturas de componentes y a JEE.

Quero Catalinas, E., García Román, A., & Peña Rodríguez, J. (2007). Mantenimiento de portales de la Información: explotación de sistemas informáticos. España: P. S.A International Thomson Editores.

Sascure CIA, L. (2012). Superintendencia de Bancos y Seguros del Ecuador.Manual de uso del Sistema de Auditorias de Prevención de Lavado de Activos (SAPLA).

SBS. (2013). Manual Técnico del Sistema de Población de Identificaciones (SPI).

Scambray , J., & Shema, M. (2002). Hacking Exposed Web Applications.

UNAM-CERT. (2009). Aspectos Básicos de la Seguridad en Aplicaciones Web.

Zalewski, M. (2012). La web enredada : guía para la seguridad de aplicaciones web modernas. Anaya Multimedia-Anaya Interactiva.

ZIMMERMAN, H. (1980). OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection. IEEE Transactions on Communications (Vol. 28(4)).